HELSINKI METROPOLIA UNIVERSITY OF APPLIED SCIENCES

Master's Degree in Information Technology

Multimedia Communications

Master's Thesis

LABORATORY SETUP FOR SESSION INITIATION PROTOCOL

Author: Korupolu Naveen Kumar
Instructor: Ville Jääskeläinen

Approved: 09.06.2011

Metropolia

PREFACE

This Master's thesis is an outcome of a big effort and great support from the Metropolia University of Applied Sciences. I would like to thank all the teachers who guided and helped me during my studies.

I would like to say many thanks to Ville Jääskeläinen, my instructor and coach. Ville Jääskeläinen is not only instructor for this thesis alone, he is the key person who motivated, guided and helped me during the whole of my studies at the university. I would also to say many thanks to Marjatta Huhta and Jonita Martelius, who guided and helped me a lot in writing this thesis.

I would like to thank my family for the unconditional support, my father Korupolu Sudhakar Reddy, my mother Korupolu Lalithamma, my wife Korupolu Swetha and my little son Korupolu Sukrith Reddy for his patience during the weekends when I was writing the text.

I would like to thank Jayasurya Kumar and all other colleagues at Wipro Technologies, who helped and motivated in completing the degree.

Helsinki, June 09, 2011
Korupolu Naveen Kumar

Metropolia

ABSTRACT

Name: Korupolu Naveen Kumar

Title: Laboratory Setup for Session Initiation Protocol

Date: June 09, 2011 **Number of pages:** 65

Degree Programme: Information Technology

Specialization: Multimedia Communications

Instructor: Ville Jääskeläinen, Senior Lecturer

This thesis is about setting up a Session Initiation Protocol laboratory in Metropolia University of Applied Sciences and designing practical lessons for the students to get more practical knowledge of Session Initiation Protocol functionality.

The project included setting up the laboratory by identifying open source software and required hardware. There are several open source software available in the Internet. Many of the available open source software were analysed as for the suitability to laboratory requirements.

An outcome of the project the best open source software were selected for laboratory setup in Metropolia University of Applied Sciences and practical lessons for the students were designed. The laboratory is now ready for the students at Metropolia to be used for their practical lessons.

Key words: Learning environment, Laboratory, SIP.

TABLE OF CONTENTS

PREFACE

ABSTRACT

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

HTTP	Hyper Text Transfer Protocol
IETF	Internet Engineering Taskforce
IM	Instant Messaging
IMPP	Instant Messaging and Presence
IP	Internet Protocol
NAT	Network Address Translation
RFC	Request for Comments
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SMS	Short Messaging Service
TCP	Transmission Control protocol
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram protocol
URI	Uniform Resource Identifier
VoIP	Voice over Internet Protocol
XMPP	Extensible Messaging and Presence Protocol

1 INTRODUCTION

This study deals with setting up a laboratory for Session Initiation Protocol at the Metropolia University of Applied Sciences. This section begins with an introduction to the Session Initiation Protocol, after that the objective of the study is clarified and the outcome of the study is described.

Session Initiation Protocol is a signalling protocol used for establishing sessions in IP (Postel 1981) networks. A session can be a simple two-way telephone call or it can be a collaborative multi-media conference session. The ability to establish these sessions means that a lot of innovative services become possible, such as voice-enriched e-commerce, web page click-to-dial, Instant Messaging with buddy lists, and IP Centrex services. SIP is based on a client server model and it is a text based protocol similar to HTTP protocol. SIP technology involves mainly two entities called User Agent Client (UAC) and SIP Server (UAS). UAC is the end system component for the call and the SIP server is the network component, which handles requests from the UAC. SIP server can work in SIP Proxy and Redirect modes depending on the network and service requirements. Currently most of the Voice over IP (VoIP) services adopted SIP as their signalling protocol. (SIPCENTER)

The Bachelor's and Master's degree programmes in ICT at Metropolia University of Applied Sciences contain only theory lessons in SIP technology. At Metropolia University of Applied Sciences practical laboratory classes are not implemented in the SIP course, because of the lack of a lab environment. The ICT lecturers argue that laboratory classes would give more practical knowledge to the students about the SIP technology. Therefore this study aimed to setup a SIP laboratory and also designed course content for the laboratory classes. The project involved identifying the required software (SIP Server and SIP Client) for the laboratory setup and designing the lessons for the laboratory classes. It involved researching various available SIP servers and SIP client software in the market and selecting the best one for the laboratory at the university. Secondly, the study involved the installation of the software and the planning of activities to be implemented using the laboratory. The approach used in this project was to experiment with the most

popular available open source SIP server software and SIP Client software. After analysing, a piece of software was to be selected to match closely with the requirements of the university.

This thesis is divided into seven sections. The first section deals with the introduction of the project and the research question. The second section describes the SIP technology and its main applications in the industry. The third section describes the learning environment to be used. The fourth section describes the hardware and software requirements for the laboratory setup. The fifth section analyses various available open source SIP server and SIP client software, and it also explains the curriculum designed for practical classes. The sixth section describes the best SIP server and client software selected for the laboratory setup. The seventh section describes what was achieved in this project and summarizes the future enhancements can be done to the laboratory and the practical lessons.

2 SESSION INITIATION PROTOCOL

This section describes the Session Initiation Protocol and its various components. Session Initiation Protocol is originally defined in RFC 3261 (Rosenberg 2002) by Internet Engineering Task Force.

SIP is request and response based protocol and it is very similar to the Hyper Text Transfer Protocol (HTTP) (Fielding et al. 1999). SIP is also text based protocol and it works on top of both Transmission Control Protocol (TCP) (Socolofsky et al. 1991) and User Datagram Protocol (UDP) (Postel 1980). SIP works on application layer and it uses the underlying IP network to carry all messages and media (like voice). SIP messages consist of headers followed by a message body. The Content-Type header in the message determines the content present in the message body.

In VoIP domain SIP protocol is mainly used along with Session Description Protocol (SDP) (Handley et al. 2006) for establishing multimedia sessions. In Instant Messaging and Presence domain, SIP protocol is used to carry present information. RFC 3261 explains the functionality of SIP UAC (Rosenberg 2002:71) and UAS (Rosenberg 2002:70). After that, some extensions for SIP protocol are also defined by IETF such as SIP/SIMPLE (Roach 2002)

Various components involved in Session Initiation Protocol are called SIP entities which are described in the sections below.

2.1 SIP ENTITIES

This section briefly describes various SIP entities and their functionalities. SIP ENTITIES are various components involved in Session Initiation Protocol functionality. There are mainly 5 entities USER AGENT SERVER (UAS), USER AGENT CLIENT (UAC), SIP PROXY, SIP REDIRECT SERVER and SIP REGISTER involved in SIP protocol. The following sections will explain briefly the functionalities of these entities.

USER AGENT SERVER is a SIP Entity which receives the requests from SIP clients and other SIP entities such as redirect server, proxy server, etc and provides

response to them. SIP server also hosts the applications for which behaviour is based on received requests. In IMPP domain SIP server will also generate NOTIFY message. USER AGENT SERVER is also called SIP Server. (Rosenberg 2002: 69)

USER AGENT CLIENT is a SIP entity which can initiate any of the SIP requests (OPTIONS, REGISTER, INVITE, ACK, BYE, CANCEL, PUBLISH and SUBSCRIBE). It is also called SIP CLIENT. (Rosenberg 2002: 71)

SIP PROXY is a SIP entity which acts as mediator between Client and Server. It acts as a proxy to the client. It receives the request from the client and forward the request to server based on the request. Proxy server can also be used for name mapping. That is a proxy server can question a location service and map an external SIP identity to an internal SIP identity. These proxy servers are not Firewalls they are independent servers on the internet that proxy the request on behalf of the user for various reasons. (Rosenberg 2002: 91)

SIP REDIRECT SERVER is a SIP entity which receives the requests from the clients and sends the response back to them asking to send request to another server. The address of the destination server will be mentioned in the contact header of the response. (Rosenberg 2002: 51)

SIP REGISTER is SIP entity which acts as a database for storing the contact details of the clients. It receives REGISTER request from clients and stores the information in location database. (Rosenberg 2002: 55)

SIP entities communicate each other by exchanging SIP messages between them. Below sections briefly describes about various SIP messages and their functionality.

2.2 SIP MESSAGES

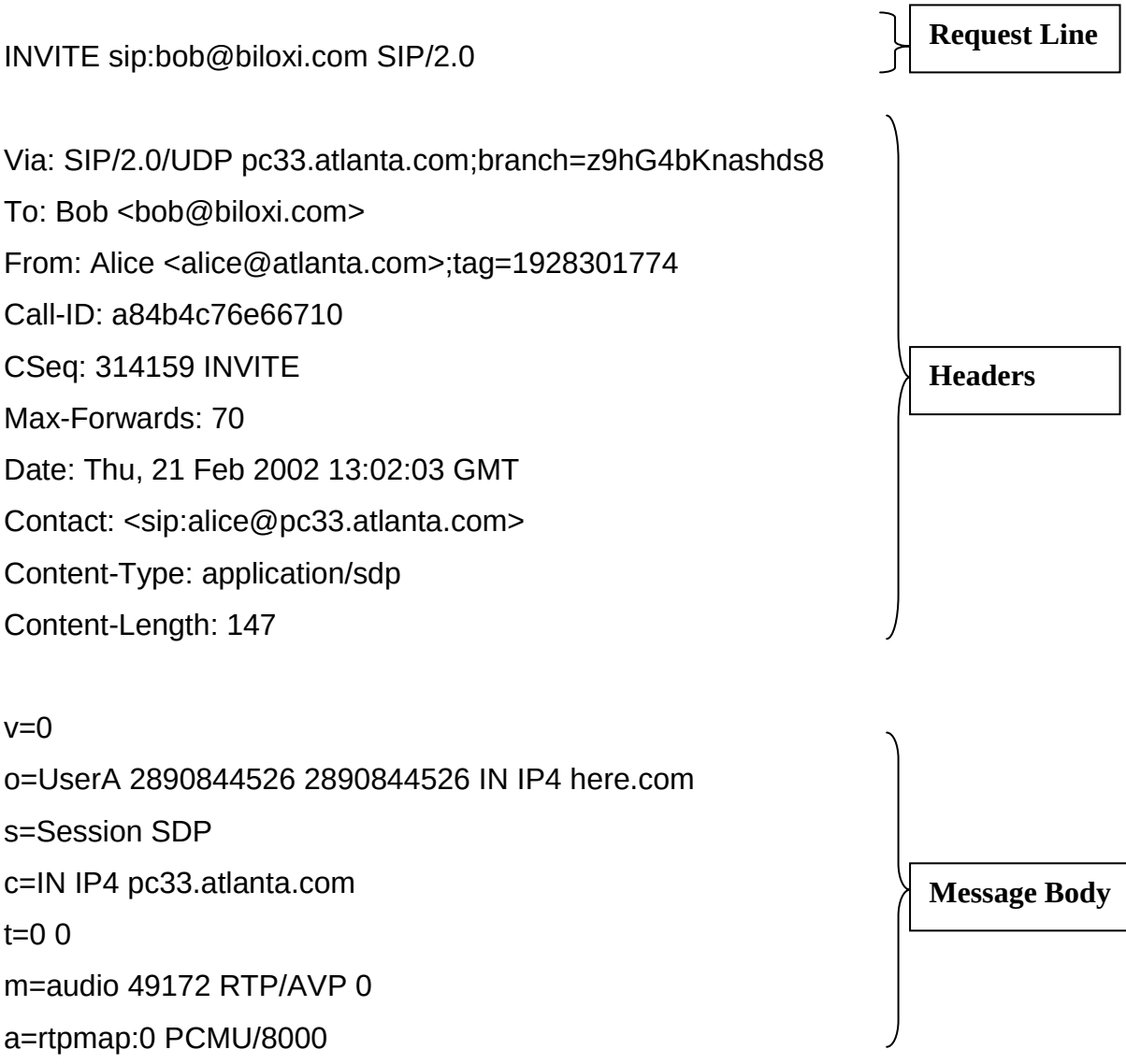
This section briefly describes briefly various SIP messages i.e. SIP Requests and SIP Responses. This section also explains the functionality of SIP messages.

SIP MESSAGES are classified as SIP Requests and SIP Responses. SIP Request are initiated by SIP client and sent to SIP Server. SIP Responses are initi-

ated by SIP server in response to request received from client and sent them to SIP clients. The following sections in this chapter describes about the general format of SIP Requests and SIP Responses. It also explains about the various SIP Requests, SIP Responses and SIP headers. (Rosenberg 2002: 26)

2.2.1 SIP REQUESTS

SIP Requests are initiated from SIP clients. SIP Request consists of request line, headers and message body as listed below. The request line includes request method, uniform recourse indicator and version. The example below explains the formation of SIP request. (Rosenberg 2002: 11)



The above example illustrates INVITE message sent by SIP client to server, to initiate the session with another SIP client. Some of the SIP requests does not contain message body such as SUBSCRIBE. A valid SIP request formulated by a UAC must, at a minimum, contain the following header fields: To, From, CSeq, Call-ID, Max-Forwards, and Via; all of these header fields are mandatory in all SIP requests. These six header fields are the fundamental building blocks of a SIP message, as they jointly provide for most of the critical message routing services including the addressing of messages, the routing of responses, limiting message propagation, ordering of messages, and the unique identification of transactions. These header fields are in addition to the mandatory request line. (Rosenberg 2002: 34) The section below briefly describe about various SIP request methods and their significance in communicating between SIP entities.

SIP REQUEST METHODS

There are mainly six SIP Request methods (mentioned below) and they are defined in primary specification (RFC 3261) for Session Initiation Protocol. The details about each of these messages will be described briefly in the following sections.

INVITE

SIP client initiates the session by sending INVITE message to the SIP server. The Request-Line has the URI, which is the next hop of the message. INVITE message example can be found below. (Rosenberg 2002: 144)

```
INVITE sip:aaa@test.com SIP/2.0

Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8
To: Bob <bbb@test.com>
From: Alice <aaa@test.com>;tag=192833455
Call-ID: a84b478ujh987
CSeq: 314159 INVITE
Max-Forwards: 10
Date: Thu, 21 Feb 2002 13:02:03 GMT
Contact: <sip:aaa@aaa.atlanta.com>
```

```
Content-Type: application/sdp
Content-Length: 147

v=0
o=UserA 2890844526 2890844526 IN IP4 here.com
s=Session SDP
c=IN IP4 pc33.atlanta.com
t=0 0
m=audio 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000

--boundary42
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s;
  handling=required
```

REGISTER

SIP client will send RIGISTER message SIP REGISTER SERVER to register its location i.e. its contact ip address and port number. REGISTER message example can be found below. (Rosenberg 2002: 212)

```
REGISTER sip:registrar.biloxi.com SIP/2.0
Via: SIP/2.0/UDP bobspc.biloxi.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Bob <sip:bob@biloxi.com>;tag=456248
Call-ID: 843817637684230@998sdasdh09
CSeq: 1826 REGISTER
Contact: <sip:bob@192.0.2.4>
Expires: 7200
Content-Length: 0
```

ACK

SIP client will send ACK message after receiving the final response from the called party for the INVITE request initiated by it self. SIP implements 3 ways handshake in establishing the session i.e. after called party accepts the session request from calling party, calling party should acknowledge the final response by sending ACK message. ACK message example can be found below. (Rosenberg 2002: 129)

ACK sip:bob@192.0.2.4 SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds9
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 ACK
Content-Length: 0

BYE

SIP client will send the BYE message to tear down the on going session. BYE message example can be found below. (Rosenberg 2002: 90)

BYE sip:alice@pc33.atlanta.com SIP/2.0
Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10
Max-Forwards: 70
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf
To: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 231 BYE
Content-Length: 0

CANCEL

CANCEL message is used to cancel the session which is not yet established i.e. the calling party can send CANCEL message before it receives response from the called party. CANCEL message example can be found below. (Rosenberg 2002: 115)

CANCEL sip:41215500309@192.168.1.15 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.9;branch=z9hG4bKfae8cb69f547b8cb
Max-Forwards: 70
To: <sip:41215500309@192.168.1.15>
From: <sip:41215500311@192.168.1.15>;tag=102
User-Agent: UDP Packet Sender
Call-ID: 070403-200101@192.168.1.9
CSeq: 5000 CANCEL
Contact: <sip:41215500311@192.168.1.9>
Content-Type: application/sdp
Content-Length: 0

OPTIONS

SIP client sends OPTIONS message to SIP server to know server's capabilities i.e. its supported methods etc. SIP server will respond with its supported capabilities to the OPTIONS request. Sample OPTIONS request and response can be found below. (Rosenberg 2002: 67)

OPTIONS sip:carol@chicago.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKhjhs8ass877
Max-Forwards: 70
To: <sip:carol@chicago.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 63104 OPTIONS
Contact: <sip:alice@pc33.atlanta.com>
Accept: application/sdp
Content-Length: 0

This section has now explained various SIP requests and their functionalities in Session Initiation Protocol. Above mentioned SIP requests are mandatory requests which should be supported by all the SIP servers.

2.2.2 SIP RESPONSES

SIP Response is a message which is initiated by the user agent server after receiving the request. This section describes briefly the various response codes and their significance.

SIP Responses contain the status line and list of headers as mentioned below.

SIPRESPONE -> [STATUS LINE] [SIP HEADERS]

STATUS LINNE -> SIP-Version SP Status-Code SP Reason-Phrase CRLF

The SIP-Version has the version of SIP protocol, for example SIP/2.0. The status code is a three digit number that indicates the response of the SIP request. Various status codes of the sip responses are explained in Table 2-1(Rosenberg 2002: 28)

STATUS CODE	DESCRIPTION
1XX	Provisional - request received, continuing to process the request
2XX	Success - the action was successfully received, understood and accepted
3XX	Redirection - further action needs to be taken in order to complete the request
4XX	Client Error - the request contains bad syntax or cannot be fulfilled at this server.
5XX	Server Error - the server failed to fulfil an apparently valid request;
6XX	Global Failure - the request cannot be fulfilled at any server.

Table 2-1 SIP Response Codes.

Table 2.1 lists all possible response codes of the SIP response. The below sections briefly describes about these response codes.

1xx Responses

These responses are provisional i.e. they gives intermediate status of the request processed. Provisional responses, also known as informational responses, indicate that the server contacted is performing some further action and does not yet have a definitive response. A server sends a 1xx response if it expects to take more than 200 milliseconds to obtain a final response. Note that 1xx responses are not transmitted reliably. They never cause the client to send an ACK. Provisional (1xx) responses may contain message bodies, including session descriptions". (Rosenberg 2002: 181).

2xx Responses

These responses are final response to the processed request. Multiple 2xx responses may arrive at the UAC for a single INVITE request due to a forking proxy. Each response is distinguished by the tag parameter in the To header field, and each represents a distinct dialog, with a distinct dialog identifier. (Rosenberg 2002: 81).

3xx Responses

A SIP client receives 3xx response when the destination is moved to some other location. A 3xx response may contain one or more Contact header field values providing new addresses where the caller might be reachable. Depending on the status code of the 3xx response, the UAC MAY choose to try those new addresses. (Rosenberg 2002: 80).

4xx Responses

A SIP client receives 4xx response when the request sent by client is invalid and server could not able to process the request. (Rosenberg 2002: 80).

5xx Responses

A SIP client receives 5xx response when server could not able to process the request, even the request is valid. (Rosenberg 2002: 80).

6xx Responses

A SIP client receives 6xx response when none of the servers can process the request. (Rosenberg 2002: 80).

Example of SIP response can be found below.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKhjhs8ass877
 ;received=192.0.2.4
To: <sip:carol@chicago.com>;tag=93810874
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 63104 OPTIONS
Contact: <sip:carol@chicago.com>
Contact: <mailto:carol@chicago.com>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE
Accept: application/sdp
Accept-Encoding: gzip
Accept-Language: en
Supported: foo
Content-Length: 0
```

Above example illustrates the “200 OK” response received from server in response to the request sent by client.

2.2.3 ***SIP HEADERS***

This section briefly describes the SIP headers which are present in a SIP request and SIP Response, they have an important role in the SIP entity functionality. SIP headers can be classified as request headers (headers used only in SIP Requests), response headers (headers used only in SIP Responses) and general headers (which are used in both SIP Requests and SIP Responses. This section briefly describes the most important headers and their functionality. (Rosenberg 2002:159).

FROM

The FROM header have the logical entity of the client which is initiating the request. It is used in both requests and responses. Example of the FROM header can be found below. (Rosenberg 2002:172).

From: Alice <sip:alice@atlanta.com>;tag=1928301774

TO

The TO header have the logical entity of the recipient to which this message is intended. It is used in both requests and responses. Example of the TO header can be found below. (Rosenberg 2002:178).

To: <sip:carol@chicago.com>

CALL-ID

The Cal-ID header have unique id, which is same for all the messages in the same dialog. UAC is responsible for generating Call-ID. Call-ID should be unique globally compared to the Call-IDs in other dialogs. Example of the Call-ID header can be found below. (Rosenberg 2002:166).

Call-ID: f81d4fae-7dec-11d0-a765-00a0c91e6bf6@foo.bar.com

CSEQ

The CSeq header has the unique id within the dialog and method name. The method name should be same as method name mentioned in the request. The CSeq value will be increased for each request within the dialog. Example of the CSeq header can be found below. (Rosenberg 2002:170).

CSeq: 4711 INVITE

CONTACT

The Contact header has the URI which can be used for contacting the particular instance of the user agent in sub sequent requests. The Contact header must contain URI for the requests which will initiate the dialog. Example of the contact header can be found below. (Rosenberg 2002:167).

Contact: <sip:alice@pc33.atlanta.com>

CONTENT-TYPE

The Content-Type header has the type of the data present in the message body. This information is very important for processing the requests by the server. Example of the Content-Type header can be found below. (Rosenberg 2002:170).

Content-Type: application/sdp

CONTENT-LENGTH

The Content-Length header has length of the data present in the message body. Example of Content-Length can be found below. (Rosenberg 2002:169).

Content-Length: 274

There are total 44 headers defined IETF specifications. (Rosenberg 2002:159). These headers have an important role in processing the SIP messages. In addition to these 44 headers, there are several other headers also defined in other SIP extension specifications such as SIP/SIMPLE (Roach 2002: 34).

2.3 SIP USE CASE

This section describes various scenarios of SIP protocol i.e. establishing session between two entities using Session Initiation Protocol. The basic SIP session setup involves a request to the SIP URL of the called end point. If the UAC knows the IP address of the UAS it will send the request directly. If not, it will use proxy server or redirect server to route the request. (Donohue et al. 2006)

2.3.1 Call Flows Between Two SIP Gateways

This section describes the SIP message flow when the proxy server is not involved in the call setup. Figure **2-1** illustrates a case when two gateways communicate with each other in order to establish a call between two analog phones behind them.

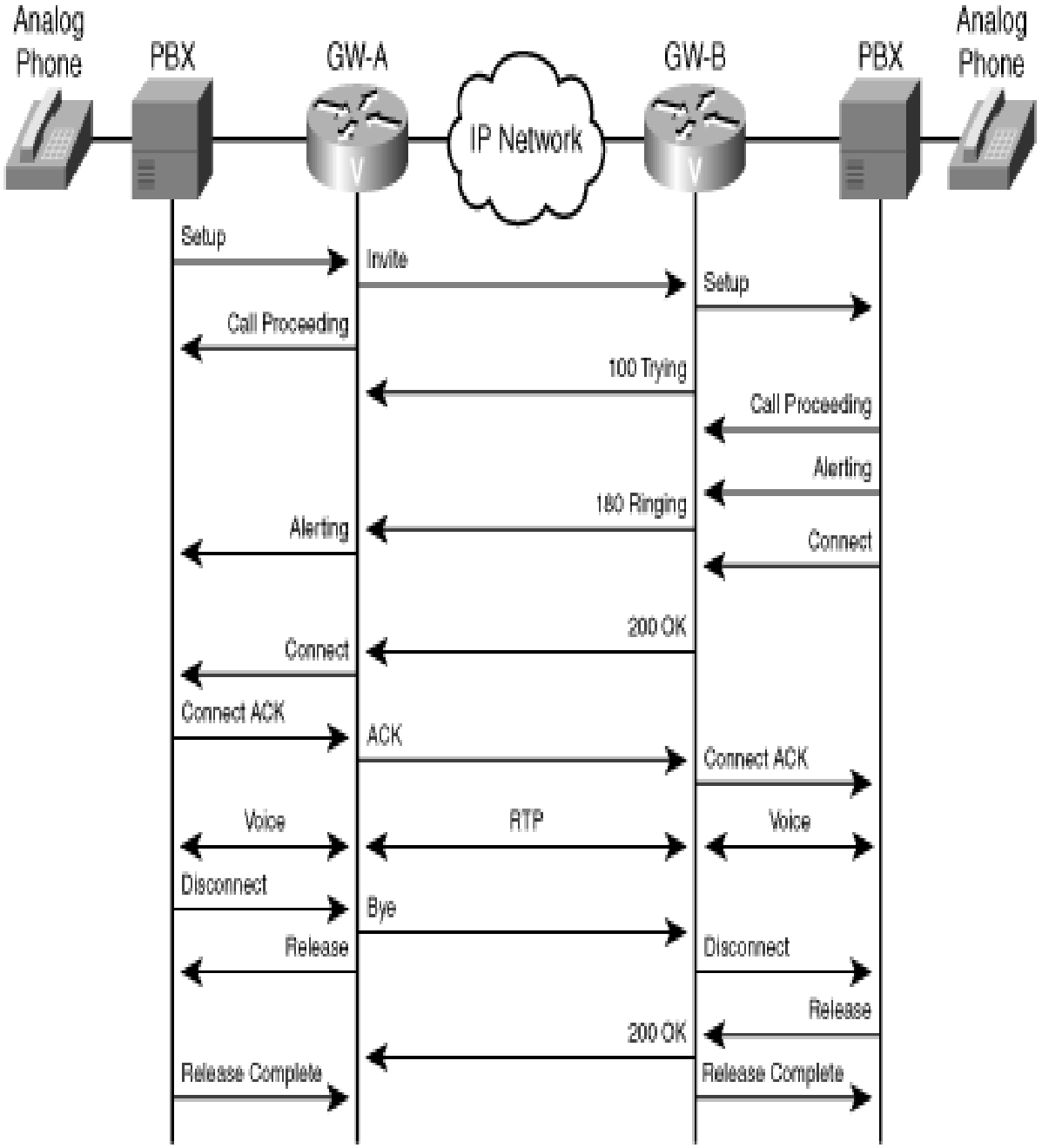


Figure 2-1 Call flow between two SIP gateways. (Donohue et al. 2006)

In Figure 2-1 (Donohue et al. 2006) the analog phone on the left initiates a call to the analog phone on the right. After the first phone initiates the call, the call flow proceeds as described below.

First the PBX sends a call setup signal to GW-A, which then sends a SIP INVITE message to GW-B. This INVITE contains SDP information for capabilities negotiation. GW-A also sends a Call Proceeding message to the PBX. Secondly GW-B exchanges call setup message with its PBX and sends SIP responses 100 (Trying) and 180 (Ringing) to GW-A. Thirdly GW-A translates these messages into analog signalling messages for its PBX. After the user on the right picks up the call, his PBX sends a Connect message to GW-B, which then forwards a SIP 200 (OK) response to GW-A. This OK response contains SDP information with the capabilities that both devices support. After that GW-A delivers a Connect message to its PBX. When the PBX acknowledges that with a Connect ACK, it sends a SIP ACK message to GW-B. Later GW-B sends a Connect acknowledgement to its PBX, and the call is active. At this point, normal voice steams exist between the two phones and the gateways, and RTP voice streams exist between the two gateways. After user on the left hangs up the phone. His PBX sends a Call Disconnect message to GW-A. GW-A then sends a SIP BYE message to GW-B and a Release message to the PBX. The PBX responds with a Release Complete message. After that GW-B sends a Call Disconnect message to its PBX, which responds with a Release message. Finally GW-B forwards a SIP 200 (OK) response to GW-A and a Release Complete message to its PBX. The call is now completely terminated. (Donohue et al. 2006)

2.3.2 Call Flow Using Proxy Server

This section describes the SIP message flow when a proxy is involved in the call setup. Figure **2-2** illustrates a case where a SIP endpoint places a call using a proxy server. The figure shows several types of endpoints:

- A PC and a PDA running a SIP application
- A SIP phone

- A cell phone that uses SIP

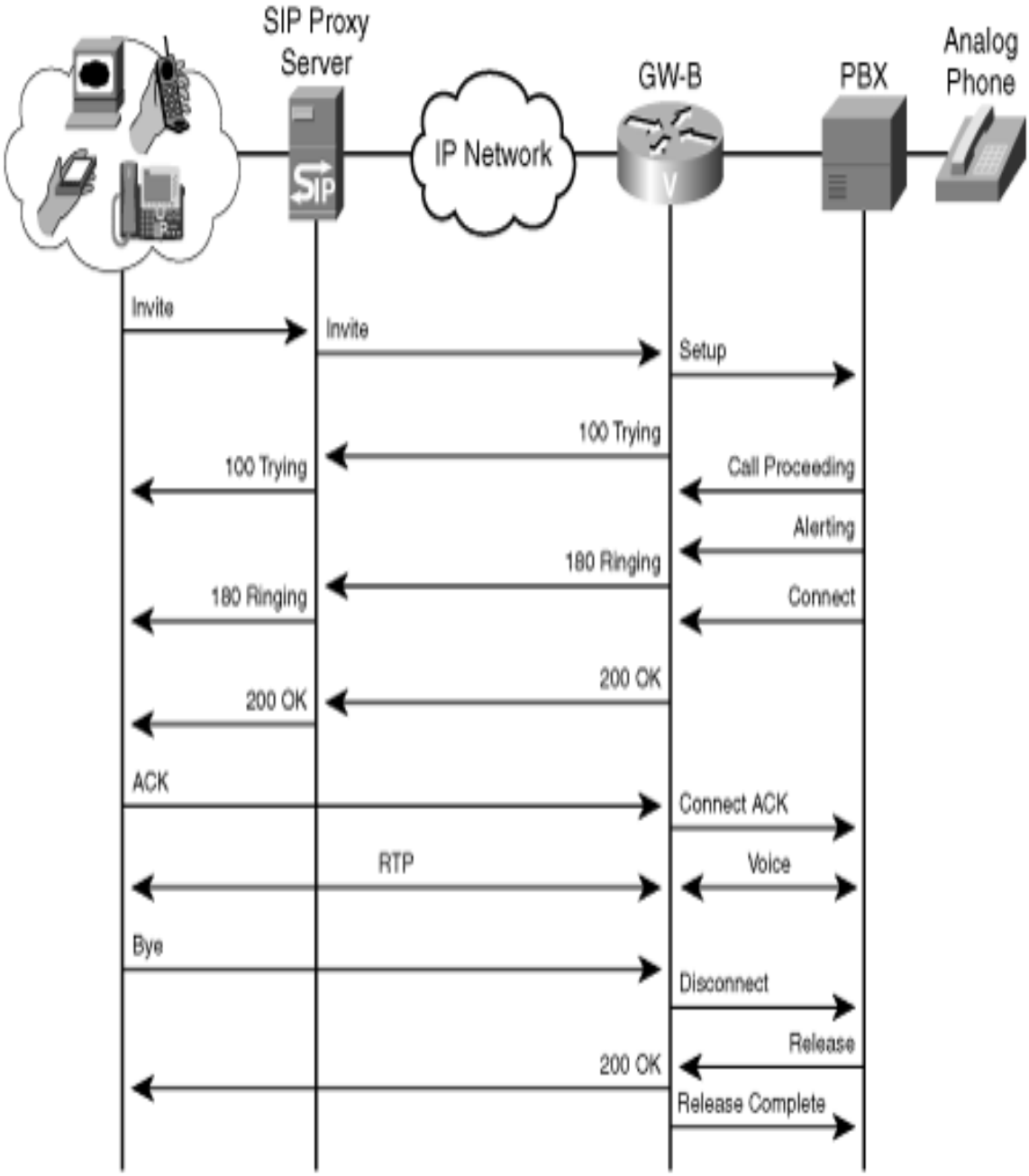


Figure 2-2 Call flow using proxy server. (Donohue et al. 2006)

In Figure 2.2 (Donohue et al. 2006) one of these endpoints places a call to an analog phone behind SIP gateway GW-B. The call flow proceeds as described below.

Firstly, the UAC sends an INVITE to its proxy server. In this INVITE, the Request-URI field contains the address of the called phone number as part of the SIP address. SDP information is included with this INVITE. Secondly, the proxy server creates a new INVITE, copying the information from the old INVITE, but replacing the Request-URI with the address of GW-B—the UAS. After GW-B receives the INVITE, it initiates a call setup with the PBX. It sends a SIP response 100 (Trying) to the proxy server which, in this example, sends a 100 response to the SIP UAC. The proxy server is not required to send this response. After that PBX sets up an analog call with the end user and sends call progress messages to GW-B. When GW-B receives the Alerting message, it sends a SIP 180 (Ringing) message to the proxy server. The proxy server sends the same message to the UAC. After the end user picks up the phone, the PBX sends a Connect message to GW-B. GW-B then sends a SIP 200 (OK) response to the proxy server, which sends it to the UAC. SDP information for the remote end is included in this OK response. The proxy server is not configured to be stateful—that is, Record Route is disabled. Therefore, the proxy server leaves the signalling path, and all further SIP signalling is directly between the UAC and GW-B. After the SIP UAC acknowledges the OK response, and a two-way RTP stream is established between the UAC and GW-B, the UAS. A two-way voice stream is established between GW-B and the PBX. Finally the UAC hangs up, it exchanges SIP BYE and OK signals with GW-B. GW-B terminates the call with the PBX. (Donohue et al. 2006).

The above two sections described the responsibility of the SIP protocol in a call setup between two clients with proxy and without proxy. The above scenarios are most common SIP use cases in the VoIP technology.

3 LEARNING ENVIRONMENTS

Learning environment can be defined in many different ways. A learning environment is a place where people can draw upon resources to make sense out of things and construct meaningful solutions to problems (Wilson 1996:3).

This study includes designing the course contents of SIP laboratory work for Metropolia University of Applied Sciences students. The aim of the course should be such that after completing the course the student would be familiar with practical knowledge of the SIP protocol and its use in VoIP and mobile domain. The below mentioned two learning environments are considered for teaching practical lessons.

3.1 Physical Environment

Physical learning environment is a place where education is arranged. It can be a classroom or a laboratory for example. It is possible that SIP practical classes can be conducted either in a class room at the same time with theoretical lessons or separately in a laboratory. In order to perform laboratory exercises both server and client machines should be connected to the same network and all the machines should be reachable to each other. One should also make sure that there are no security restriction in the network which impacts communication between client and server.

In order to conduct practical lessons in the classroom, a SIP server should be installed in the laptop and configured for the practical lessons. The teacher should bring the laptop (SIP server) to the class room and it should be connected to a separate WLAN network in the class room. Students can install the SIP client software and they can connect to the SIP server using the same WLAN in order to complete the practical exercises. Using this mode it is possible that all the students can perform practical exercises simultaneously.

In order to conduct practical lessons in the laboratory, all the client and server machines should be connected to same network. Students can go to the laboratory in groups and perform practical exercises. This mode of environment is not suitable if too large number of students want to perform the exercises at the same time.

3.2 Social Environment

In a Social environment students can perform laboratory work in groups and they can share the knowledge.

Current brain science research tells us that the brain is social. The second brain/mind principle of the Caines speaks directly to the social nature of the brain: "It is now clear that throughout our lives, our brain/minds change in response to their engagement with others so much so that individuals must always be seen to be integral parts of larger social systems. Indeed, part of our identity depends on establishing community and finding ways to belong." (Bell Annette)

A safe social environment for students requires a strong sense of community. New learning involves taking risks. Even if a student is highly motivated and has a high self concept, without a supportive community, he or she will be unwilling to take risks, and without creative risk taking, deep learning is not going to happen. (Bell Annette).

SIP practical exercises should be done in groups of students rather than individually for better understanding of the SIP protocol functionality. As practical lessons involve multiple clients, it is suitable for the students to perform exercises in groups. This study concentrates on both the physical and social learning environment implementation to teach Session Initiation Protocol.

4 LABORATORY

This aim of this research study was to set up the SIP laboratory in Metropolia University of Applied Sciences along with the course design for SIP protocol. This chapter describes the hardware, software and budget requirements for the laboratory setup.

HARDWARE REQUIREMENTS

Based on the requirements given by Metropolia University of Applied Sciences the laboratory setup requires the below mentioned minimum hardware.

- One high end computer to run the SIP server software.
- Two or more computers to run SIP client software. These machines should also contain sound card installed in it.
- Two or more headsets connected to client machines.

The existing laboratory machines can be used for running SIP client software. Both the SIP server machines and client machines should be in the same network.

SOFTWARE REQUIREMENTS

The required software for building the SIP laboratory in Metropolia University of Applied Sciences are listed in Table **4**-**1**.

Software	Purpose
Ubuntu operating system.	Ubuntu operating system required on SIP server machine.
OpenSIPS SIP Server	OpenSIPS SIP server software contains most of the SIP protocol functionalities.
Linphone SIP Client	Linphone SIP client software contains most of the SIP client functionalities including SIP/SIMPLE.
MySql	MySql database is as database of SIP server.
Wireshark	Wireshark protocol analyser for analyzing the messages exchanged between SIP client and SIP server.

Table 4-1 Software requirement for laboratory setup

The purpose of each selected software is also mentioned in Table 4.1. These are the minimum software requirements for the SIP laboratory setup. The software mentioned in table 4.1 were selected after the analysis of many open source software, as explained in section 5.2.

BUDGET

There is no special budget allocated for the lab setup. Metropolia University of Applied Sciences will provide the required hardware for the lab setup. Most of the available lab equipment can be used in the new laboratory setup.

5 METHODS AND MATERIAL

This section describes the methods and criteria used in identifying the SIP server, SIP client software and laboratory exercises.

5.1 Method and Process

After discussion with the teacher at Metropolia University of Applied Sciences, the below mentioned minimal requirements were identified for selecting SIP server software to the laboratory.

- Should be open source software (i.e. there is no licensing requirement involved in this).
- Easy to install.
- Easy to configure, should support all the basic SIP functionalities.
- Should be able to handle multiple client requests simultaneously.
- It should be reliable i.e. software should be stable.
- It should support SIP/SIMPLE support.

With the SIP server software satisfying the above requirements, students can test all the basis SIP functionalities in addition to all the practical lessons designed as part of this study.

After discussions with the teacher responsible for the course at Metropolia University of Applied Sciences, the below mentioned minimal requirements were identified for selecting SIP client software to the laboratory

- Should be open source software (i.e. there is no licensing requirement involved in this).
- Easy to install.
- Easy to configure.

- Should support all the basic SIP functionalities.
- Should be reliable i.e. software should be stable.
- Students can observe the ongoing SIP transactions.
- Software should be available in Windows platform

With the SIP client software satisfying the above requirements, students can test the basis SIP functionalities in addition to all the practical lessons designed as part of this study.

5.2 Experiment

There are many open source SIP server and client software available in the Internet. (OPEN SOURCE SOFTWARE). All the available server and client software were analyzed for their suitability as the laboratory requirements.

Many of the open source SIP servers available in the market implemented only basic SIP protocol functionality mentioned in RFC-3261, for example MjSip server (MJSIP) and OpenSIPStack server (OPENSIPSTACK). One of the requirements for selecting the server software was that it should support SIP/SIMPLE functionality, where as these software do not support SIP/SIMPLE. It was also observed that many software are not easy to configure. Open source community is not very much active in these projects. Many of these software do not have the more recent features, which will prevent students from performing exercises on the latest features.

Many of the open source SIP client software are available in the Internet, for example Office SIP Messenger (OFFICEMESSANGER), Jisti Phone (JISTIPHONE), etc. In this project only Windows based clients were considered for the laboratory as most of the machines in the university laboratory contain Windows operating system. One of the motives of the present study was to reuse the existing hardware in the laboratory. There were no other Windows based clients found containing UI and supporting the SIP/SIMPLE functionality. Only Linphone (LINPHONE) software had all the required features.

After analysis the below mentioned SIP server and Client software were selected for the laboratory setup.

- OpenSIPS SIP server. (OPENSIPS)
- Linphone SIP Client. (LINPHONE)

The two are briefly described in this section. OpenSIPS (Open SIP Server) is a mature *Open Source* implementation of a SIP server. OpenSIPS is more than a SIP proxy/router as it includes application-level functionalities. OpenSIPS, as a SIP server, is the core component of any SIP-based VoIP solution. With a very flexible and customizable routing engine, OpenSIPS 'unifies voice, video, IM and presence services in a highly efficient way, thanks to its scalable (modular) design. What OpenSIPS has to offer, comes in a reliable and high-performance flavour - OpenSIPS is one of the fastest SIP servers, with a throughput that confirms it as a solution up to enterprise or carrier-grade class. (OPENSIPS). OpenSIPS server also supports SIP/SIMPLE functionality apart from basic SIP functionalities mentioned in RFC-3261. OpenSIPS software is easy to install and configure for laboratory purpose. Many professionals are actively involving in OpenSIPS community which helps in supporting the OpenSIPS server maintenance and update.

Whereas, Linphone is an open source SIP client software, which was used to communicate with SIP server. With Linphone one can communicate freely with people over the Internet, with voice, video, and text instant messaging. Linphone makes use of the SIP protocol, an open standard for internet telephony. One can use Linphone with any SIP VoIP operator, including our free SIP audio/video service. Linphone is free-software (or open-source), one can download and redistribute it freely. Linphone is available for desktop computers: Linux, Windows, MacOSX, and for mobile phones: Android, iPhone, Blackberry. (LINPHONE). Linphone UI is very user friendly and easy to configure for the laboratory purpose. Linphone is one of the best open source SIP clients available in the industry. It has SIP/SIMPLE support which was one of the main requirements. Using Linphone debug functionality students can see all the messages exchanged between Linphone and the SIP server, which gives more practical knowledge of the SIP protocol for the students.

6 RESULTS AND ANALYSIS

The outcome of this research study was to build a laboratory (SIP environment) in Metropolia University of Applied Sciences and also to design the curriculum for the SIP laboratory classes. This section describes the laboratory setup implemented in Metropolia University of Applied Sciences and the practical lessons designed for the students.

6.1 LABORATORY SETUP

The laboratory setup required one high end machine for installing the SIP server software and at least two machines for installing the SIP client software and Wire-shark protocol analyser. Wireshark protocol analyser is used for monitoring the messages between client and server. Figure **6-1** illustrates the recommended laboratory setup for performing the laboratory exercise designed in this study.

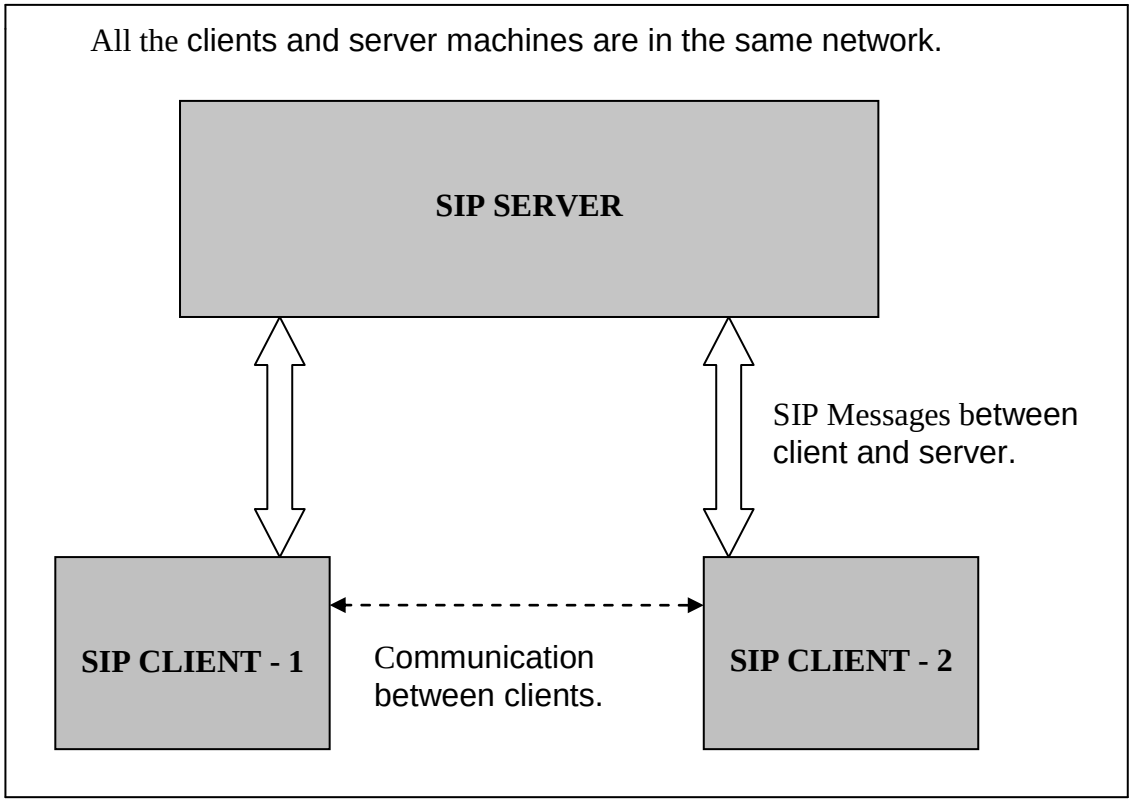


Figure 6-1 SIP Laboratory setup in University.

In order to perform the SIP laboratory exercises both server and client machines should be in the same network as in Figure **6-1**. The network should not contain

any security restrictions, which impacts communication between client and server i.e. dropping the TCP and UDP packets from unknown hosts, etc. It is recommended to do practical lessons in a private network, for example the entire client and server machines connected to a single hub or in any network which does not contain any security restrictions. It is advisable not to do practical lessons by connecting server and client machines in the university official network, since this network may contains several security restrictions which may impact the communication between clients and server machines.

By using the selected SIP server and SIP client software students can perform all the identified practical lessons in the university laboratory. The laboratory was successfully demonstrated to the Metropolia University of Applied Sciences teachers by performing all the practical lessons designed in this study.

6.2 LABORATORY EXERCISES

This section describes some practical exercises to be completed by each student in order to understand the very basic SIP protocol functionality. Each designed exercise is targeted to understand the very basic functionality of the protocol. Students can perform other exercises also by using the existing laboratory setup. All the exercises mentioned in this section are created and tested by the author in the university SIP laboratory. At the end of each exercise a set of questionnaires and more exercises were presented to the students, these questions and exercises were aimed for students to gain more knowledge in SIP functionality. All the figures mentioned in the chapter below illustrate the different steps needed to perform the intended exercises.

Exercise 1

After completing this exercise a student should be able to understand the SIP REGISTER message and its functionality.

1. Make sure that the SIP server is up and running. Refer to Appendix B to check the server status.

2. Make sure that the SIP clients (user1 and user2) are configured. Refer to Appendix D for configuring SIP client.

3. Test that the clients are able to communicate with the server by using ping command from Windows Command Prompt.

4. Start the Wireshark protocol analyser to capture the packets between SIP clients and server.

5. Launch the SIP client (Linphone) application which is configured for user 1.

6. If the SIP Client asks for the password, please provide the password configured for the user1 in the server and click OK button. See Figure **6-2** for the appearance of the password form.

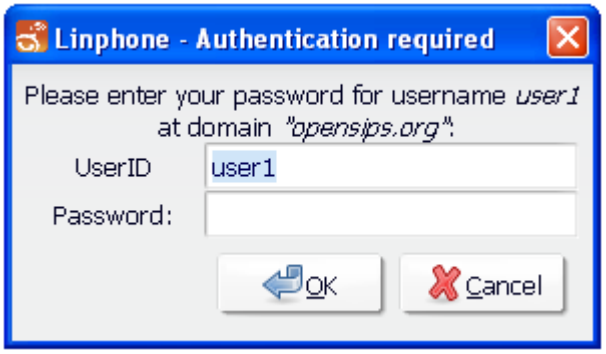


Figure 6-2 Password form

7. The SIP client should be successfully registered at SIP server and the same in displayed in Linphone UI as illustrated in Figure **6-3**.

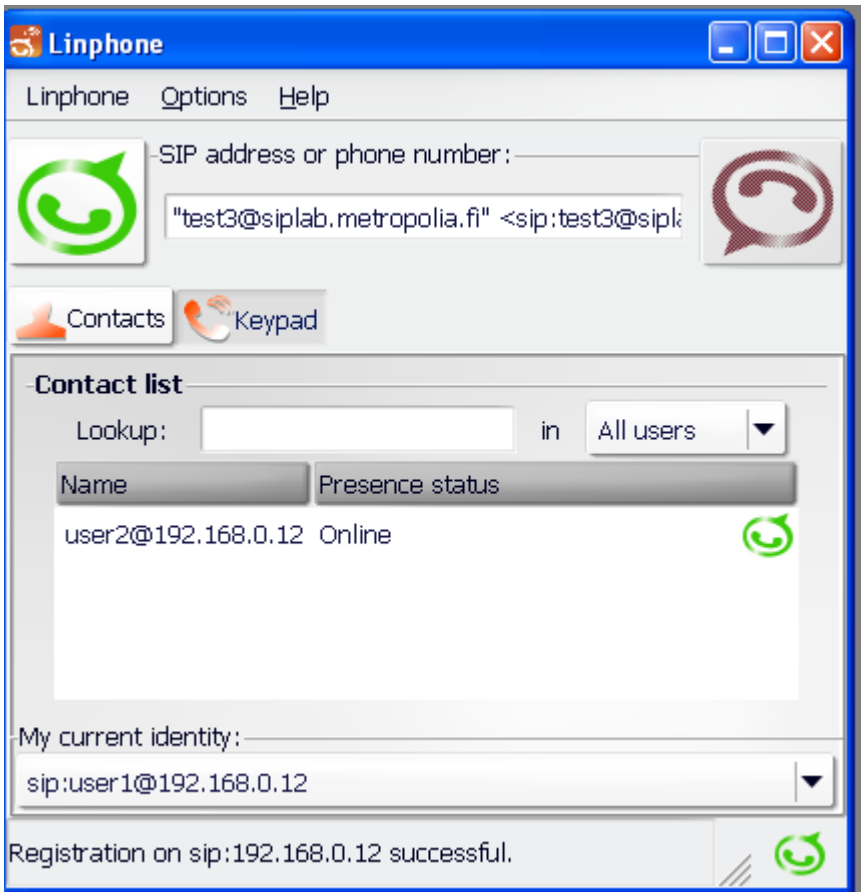


Figure 6-3 Linphone UI after successful registration

8. Stop the Wireshark protocol analyzer and observe the messages exchanged between SIP client and server.

Questions

1. How many REGISTER messages are exchanged between the client and server? If more than one REGISTER messages are exchanged, explain the reason for the same?
2. Check whether REGISTER messages will succeed if a wrong password provided at step 6.

Exercise 2

After completing this exercise the student should be able to understand the SIP message flow between the SIP client and server when a voice call is established between two SIP clients.

1. Make sure that the SIP server is up and running. Refer to Appendix B to check the server status.

2. Make sure that the SIP clients (user1 and user2) are configured. Refer to Appendix D for configuring the SIP client.

3. Make sure that both SIP clients are started and they are successfully registered with the server.

4. Make sure that the other contact presence information is online. In Figure **6-4** below, user2 is displayed as online in user1 contact list.

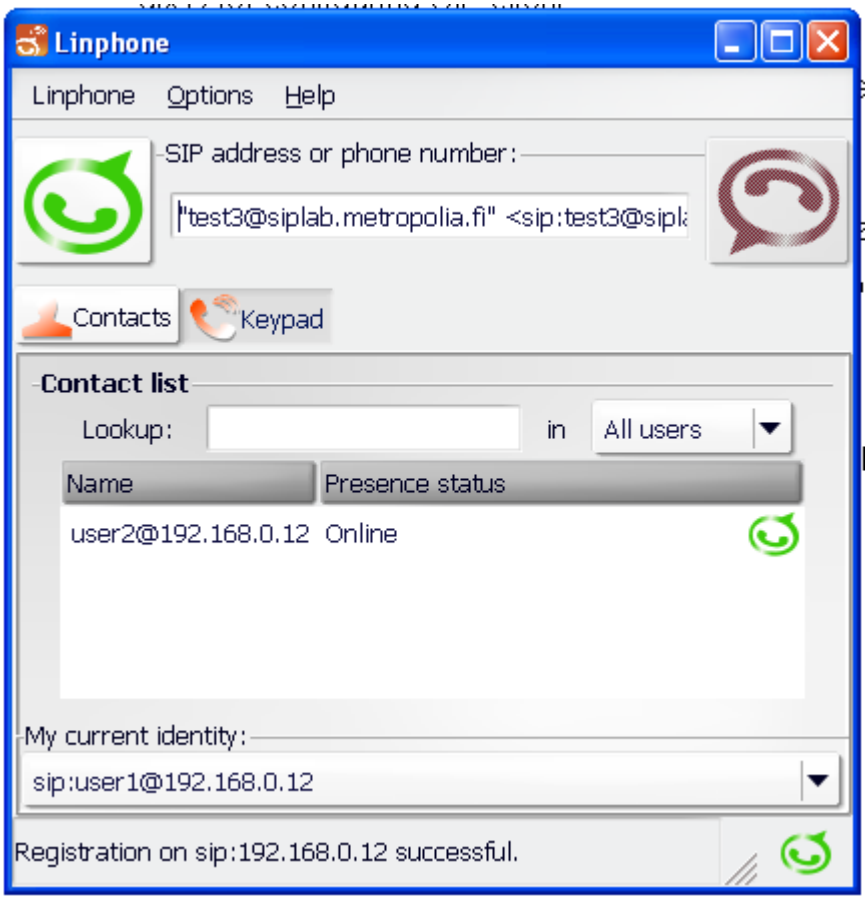


Figure 6-4 Linphone UI with user online status

5. Start the Wireshark protocol analyser to capture the packets between the SIP clients and server.

6. Initiate a voice call between two SIP clients by clicking on green button located on top left corner of the Linphone UI. In Figure **6-5** below, user 1 is calling user2.

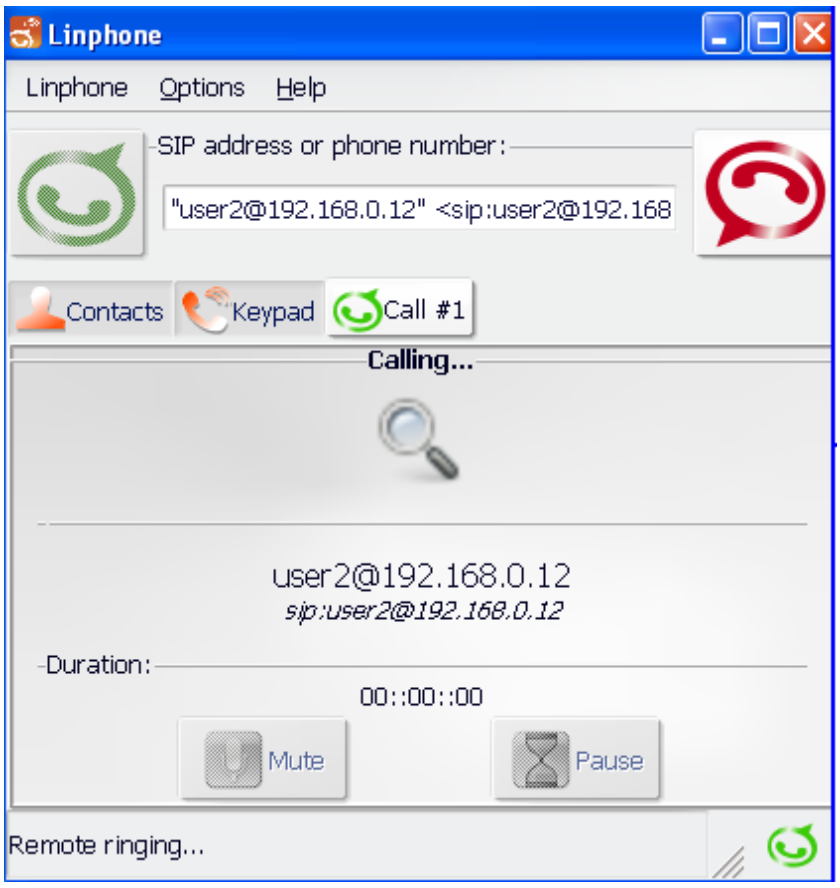


Figure 6-5 Linphone UI when a call initiated.

7. Recipient accepts the call i.e. user 2 accept the call. Figure **6-6** below illustrates the Linphone UI after a voice call established between user1 and user2.

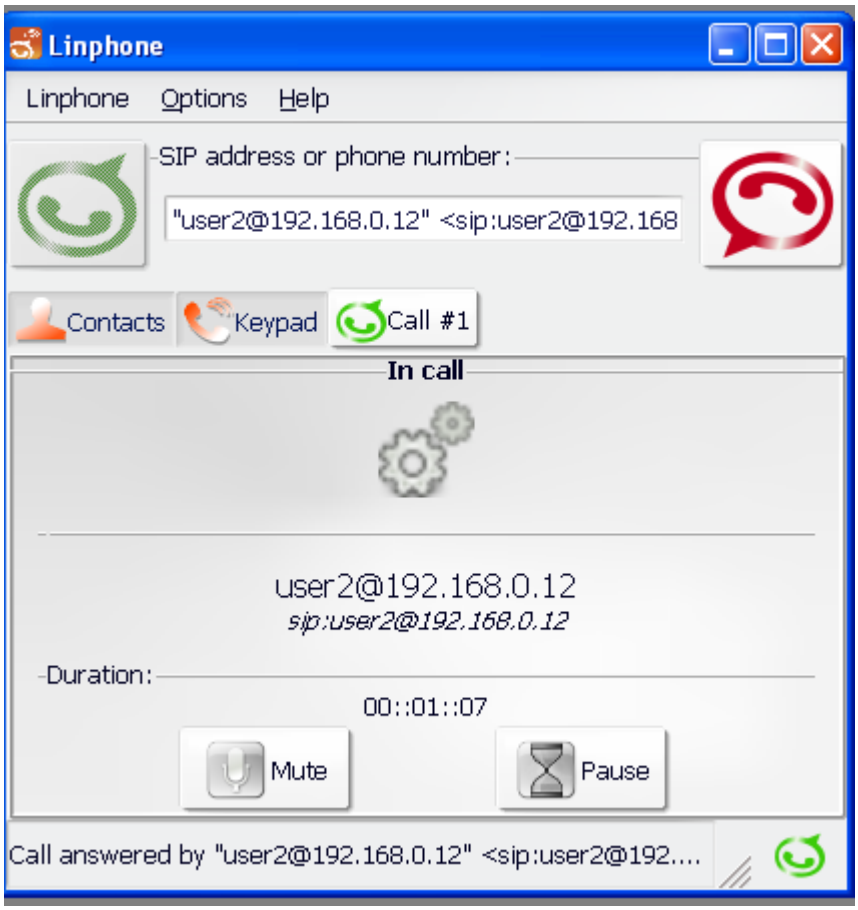


Figure 6-6 Linphone UI when a call established

8. Disconnect the call i.e. user 1 closes the call.

9. Stop the Wireshark protocol analyser and analyse the messages exchanged between the sip clients and server.

Questions and more exercises

1. How many messages were exchanged between the call initiator (i.e. user1) and the server?
2. List down all the messages exchanged between the call initiator and the server during the call process and explain the significance of the each message.
3. How many messages were exchanged between the call receiver (i.e. user 2) and the server?
4. List down all the messages were exchanged between the call receiver and the server during the call process and explain the significance of each message.

Exercise 3

After completing this exercise the student should be able to understand the SIP/SIMPLE functionality.

1. Make sure that the SIP server is up and running. Refer to Appendix B to check the server status.

2. Make sure that the SIP clients (user1 and user2) are configured. Refer to Appendix D for configuring SIP client.

3. Start the Wireshark protocol analyser to capture the packets between SIP clients and server.

4. Make sure that both SIP clients are started and they are successfully registered with the server.

5. Make sure that user2 is in the contact list of user1 and his presence status is online. During the start up user1 should send SUBSCRIBE message to the server for user2 presence information and user1 should receive the user2 presence information i.e. user2 is online. The same should be displayed in user1 contact list as shown in Figure **6**-**7** . In Figure **6**-**7** below user2 presence status is displayed as online in user1 contact list.

Figure 6-7 Linphone UI with other user presence status online

6. Change the status of user2 as Busy. The updated status of user2 should be displayed in user1 contact list as illustrated below in Figure **6**-**8**

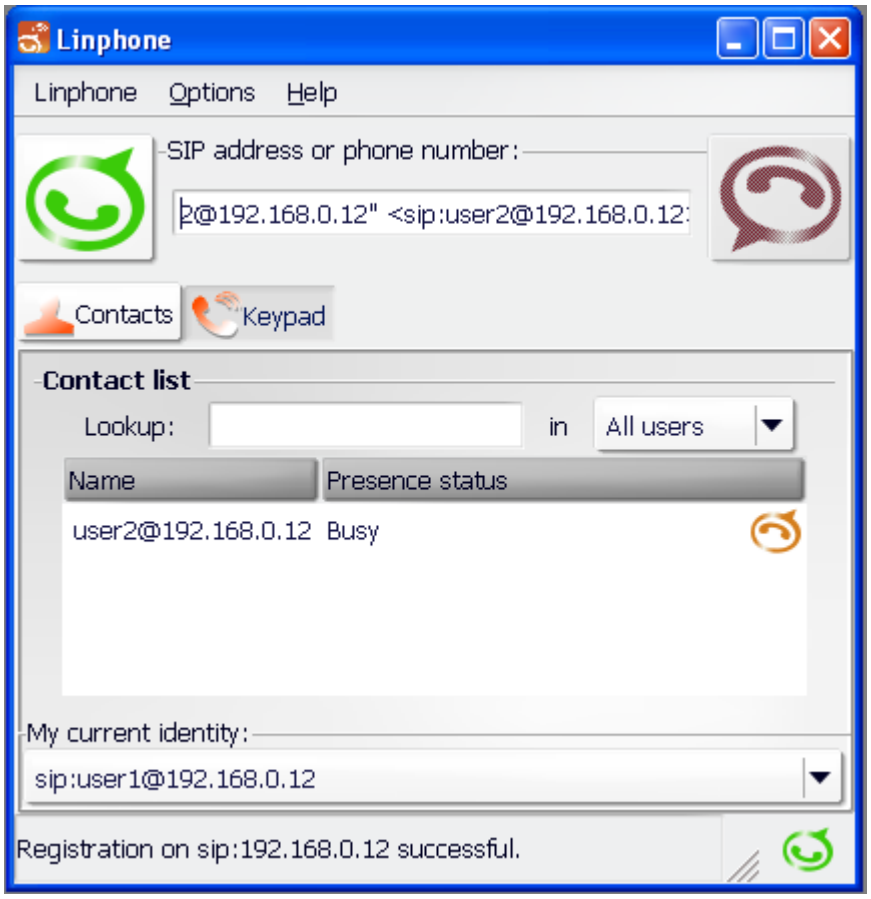


Figure 6-8 Linphone UI with other user presence status as busy

7. Again change the presence status of user2 as Online. The updated status of user2 should be displayed in user1 contact list as online

8. Stop the Wireshark protocol analyzer and observe the messages exchanged between SIP client and server.

Questions and more exercises

1. What message user1 received from the server when user2 changed his status?
2. What message user2 sent to the server when user2 changes its status?
3. Change user1 client status and observe the behaviour of user1 status in user2 contact list. Analyse what messages are exchanged between clients and server?

7 DISCUSSION AND CONCLUSIONS

The aim of the research was to identify the required software for setting up the SIP laboratory in Metropolia University of Applied Sciences and to configure the laboratory setup. Also practical lessons were designed for the students in order for them to understand the SIP protocol functionality.

The laboratory was designed to be used by the university students and they can perform the identified exercises to understand the functionality of the SIP. The planned practical lessons were demonstrated to the university teachers in the laboratory and they were fairly pleased with the outcome. With the current laboratory setup students can experiment any basic SIP functionality in addition to the practical lessons designed in this study.

Currently planned practical lessons required only one SIP server, which is acting as both SIP Server and Proxy. In future the laboratory environment can be extended by using multiple SIP servers and multiple domains support. The practical lessons can be updated to match the new laboratory setup.

In the current laboratory setup clients are able to reach the server directly i.e. without involving NAT. In future more exercises can be designed so that NAT can be incorporated between clients and server machines. Also exercises can be designed by applying security restrictions in the network which will impact communication between clients and servers.

REFERENCES

[1]. Bell Annette, Dialogue on Learning, [WWW document] http://www.dialogueonlearning.tc3.edu/model/environment/Introduction-grp.htm (Accessed Apr 10, 2011)

[2]. Donohue, D., Mallory, D. and Salhoff, K. (2006). Session Initiation Protocol. http://www.ciscopress.com/articles/article.asp?p=664148&seqNum=2 (Accessed Jan 29, 2011)

[3]. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T(June 1999). IETF Standards, Hypertext Transfer Protocol -- HTTP/1.1 (HTTP), (June 1999) [WWW document] http://www.ietf.org/rfc/rfc2616.txt (Accessed Apr 08, 2011)

[4]. Handley, M.,Jacobson, V. and Perkins, C. (July 2006). IETF Standards, Session Description Protocol (SDP), (July 2006) [WWW document] http://www.ietf.org/rfc/rfc4566.txt (Accessed Apr 08, 2011)

[5]. JISTIPHONE [WWW document] http://jitsi.org// (Accessed May 25, 2011)

[6]. LINPHONE [WWW document] http://www.linphone.org/ (Accessed May 25, 2011)

[7]. MYSQL [WWW document] http://www.mysql.com/ (Accessed May 25, 2011)

[8]. MJSIP [WWW document] http://www.mjsip.org/ (Accessed May 25, 2011)

[9]. OFFICEMESSANGER [WWW document] http://www.officesip.com/messenger.html (Accessed May 25, 2011)

[10]. OPENSIPS [WWW document] http://www.opensips.org/ (Accessed Jan 29, 2011)

[11]. OPENSIPSTACK [WWW document] http://www.opensipstack.org/ (Accessed May 25, 2011)

[12]. OPEN SOURCE SOFTWARE [WWW document] http://www.voipinfo.org/wiki/view/Open+Source+VOIP+Software (Accessed Jan 29, 2011)

[13]. Postel, J (September 1981). IETF Standards Internet Protocol (IP), (September 1981) [WWW document] http://www.ietf.org/rfc/rfc791.txt (Accessed Apr 08, 2011)

[14]. Postel, J (August 1980). IETF Standards, User Datagram Protocol (UDP), (August 1980) [WWW document] http://www.ietf.org/rfc/rfc768.txt (Accessed Apr 08, 2011)

[15]. Roach, A. B. (June 2002). IETF Standards, Session Initiation Protocol (SIP) - Specific Event Notification, (June 2002) [WWW document] http://www.ietf.org/rfc/rfc3265.txt (Accessed Jan 29, 2011)

[16]. Rosenberg, J (June 2002). IETF Standards, Session Initiation Protocol (SIP), (June 2002) [WWW document] http://www.ietf.org/rfc/rfc3261.txt (Accessed Jan 29, 2011)

[17]. Socolofsky, T. and Kale, C. (January 1991). IETF Standards, A TCP/IP Tutorial (TCP), (January 1991) [WWW document] http://www.ietf.org/rfc/rfc1180.txt (Accessed Apr 08, 2011)

[18]. Wilson B.G. (1996). Constructivist Learning Environments, New Jersey : Educational Technology Publications Inc.

[19]. SIPCENTER [WWW document] http://www.sipcenter.com (Accessed Jan 29, 2011)

[20]. UBUNTU [WWW document] http://www.ubuntu.com/ (Accessed May 25, 2011)

[21]. WIRESHARK [WWW document] http://www.wireshark.org/ (Accessed May 25, 2011)

APPENDIX A OPENSIPS SERVER INSTALLATION AND CONFIGURATION

This section describes the procedure for installing the OpenSIPS SIP server in laboratory. The below mentioned procedure briefly explains the procedure for installing the OpenSIPS software (OPENSIPS). ***It is recommended that always follow the latest instructions mentioned at www.opensips.org for installation and configuration.***

1. Download latest Ubuntu operations system from www.ubuntu.com and install the same in server machine. Follow the installation instructions mentioned at www.ubuntu.com

2. After successful installation of Ubuntu operating system, start the server machine and connect it to the internet.

3. Download all the debian packages required for OpenSIPS from the location http://opensips.org/pub/opensips/latest/packages/

4. Install all the downloaded packages one by one with the below command

   dpkg -i <package_name>

5. Configure the data base type as MySql in the /etc/opensipsctlrc file.

6. Create SQL tables which are required for OpenSIPS SIP server with the below script. This script will prompt for password of MYSQL "root" user.

   /usr/sbin/opensipsdbctl create

7. Configure opensips to use MYSQL database. Please refer Appendix F for the actual configuration files used in laboratory in Metropolia University of Applied Sciences.

   uncomment below mentioned lines in /etc/opensips/opensips.cfg file

   - loadmodule "/usr/lib/opensips/modules/db_mysql.so"
   - loadmodule "/usr/lib/opensips/modules/auth.so"
   - loadmodule "/usr/lib/opensips/modules/auth_db.so"
   - modparam("usrloc", "db_mode", 2)
   - modparam("auth", "calculate_ha1", yes)
   - modparam("auth_db", "password_column", "password")

8. Configure below mentioned parameters in /etc/opensips/opensipsctl file. Please refer Appendix F for the actual configuration files used in laboratory in Metropolia University of Applied Sciences.

SIP_DOMAIN: your SIP domain
DBENGINE : MYSQL
DBHOST - database host
DBNAME - database name
DBRWUSER - database read/write user
DBROUSER - database read only user
DBROPW - password for database read only user
DBROOTUSER - database super user
ALIASES_TYPE - type of aliases used:
DB - database aliases
UL - usrloc aliases
default none
CTLENGINE - control engine: FIFO or UNIXSOCK
OSIPS_FIFO - path to FIFO file

9. Restart the server with the below command.

/etc/init.d/opensips restart

APPENDIX B OPENSIPS SERVER ADMINISTRATION

This section describes briefly explains the important administration command needed frequently by the administrator (OPENSIPS). ***It is recommended that always follow the latest instructions mentioned at www.opensips.org for administration.***

1. Starting the OpenSIPS server run the below command in super user mode

   /etc/init.d/opensips start.

2. Stopping the OpenSIPS server run the below command in super user mode

   /etc/init.d/opensips stop.

3. Restarting the OpenSIPS server run the below command in super user mode

   /etc/init.d/opensips restart.

4. To check the status of the OpenSIPS server run the below command in super user mode

   /etc/init.d/opensips status

5. To check the online users execute the below command.

   /usr/sbin/opensipsctl ul show.

6. To check the server internal status execute the below command.

   /usr/sbin/opensipsctl moni

7. To add any user to the database, use the below command the below command.

   /usr/sbin/opensipsctl add <username> <password>

Ex:- /usr/sbin/opensipsctl add test3 test3

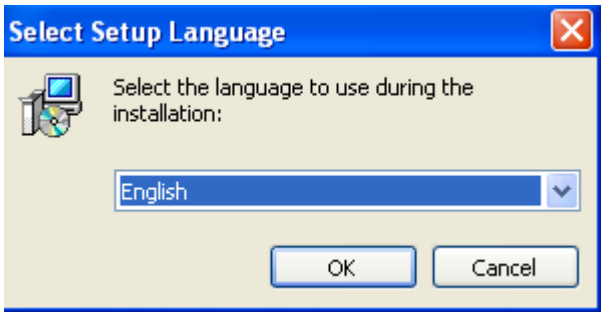
APPENDIX C LINPHONE SIP CLIENT INSTALLATION

This section briefly describes about the installation procedure of Linphone SIP client. The below mentioned procedure briefly explains step by step installation procedure for installing the Linphone SIP client software in WINDOWS operating system (LINPHONE). ***It is recommended that always follow the latest instructions mentioned at www.linphone.org for installing the Linphone SIP client software.***

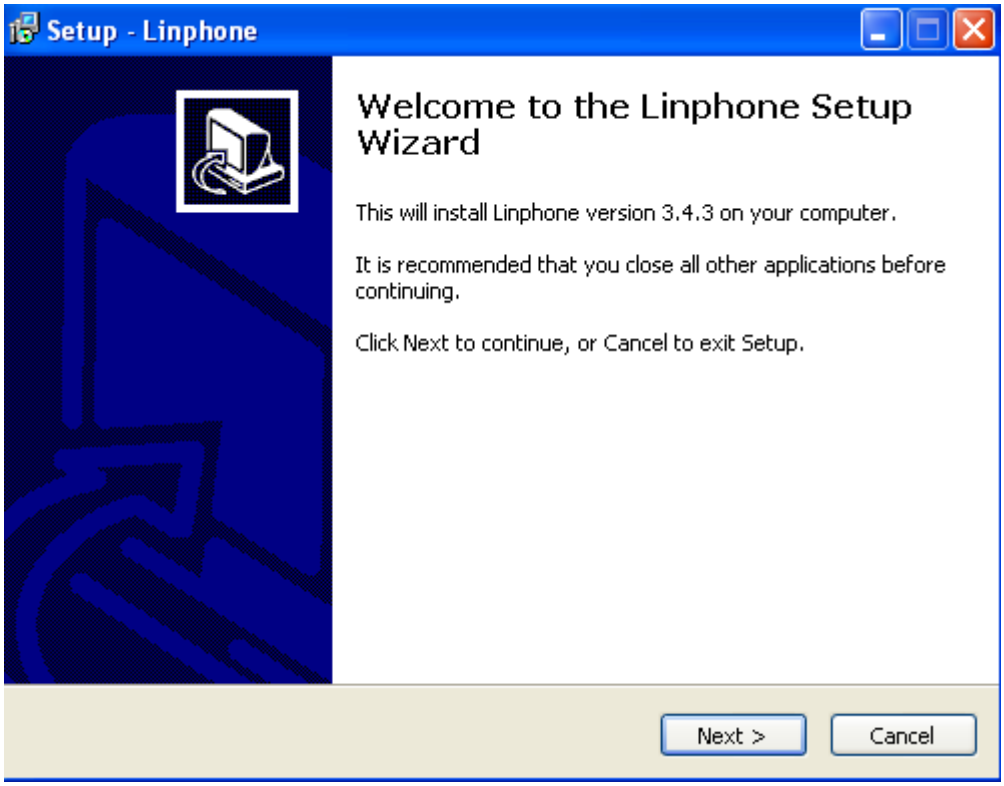
1. Download latest Linphone SIP client software from www.linphone.org

2. Double click on the downloaded installer file.

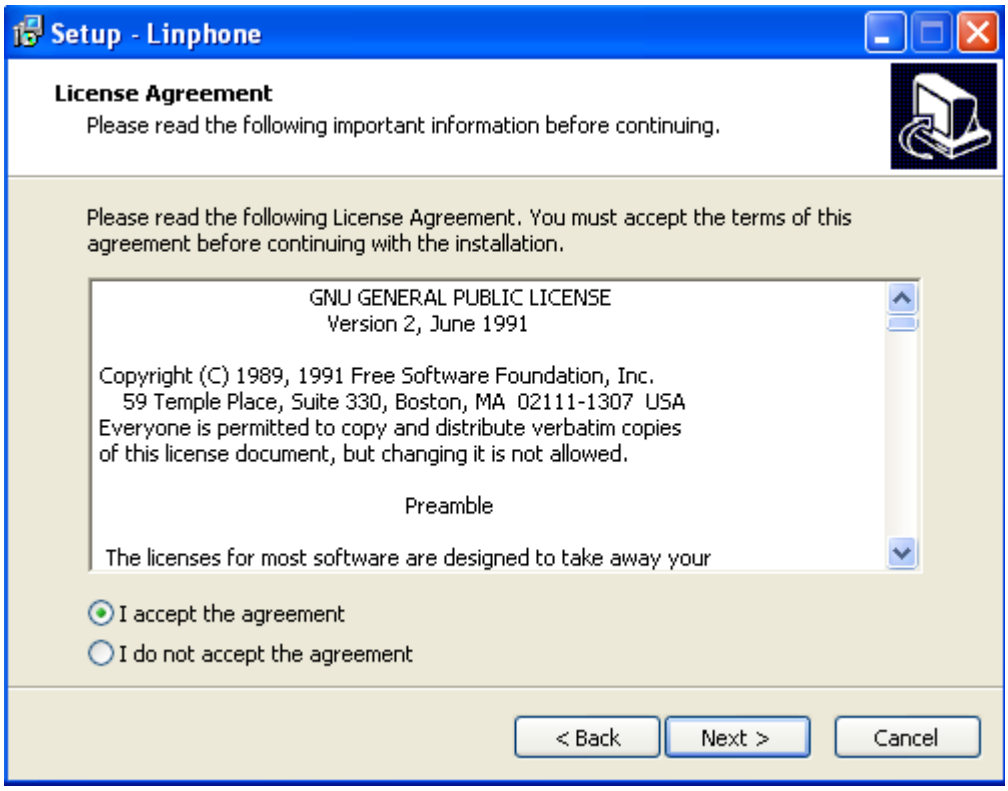
3. Tap on Run button.



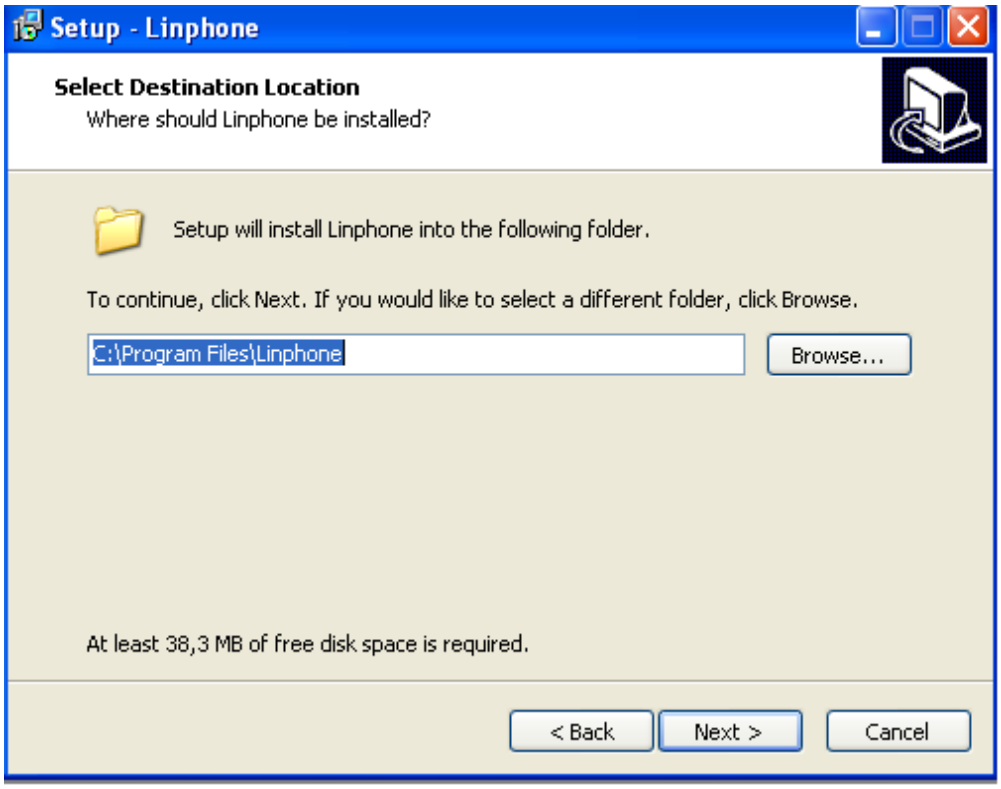
4. Tap on OK button.



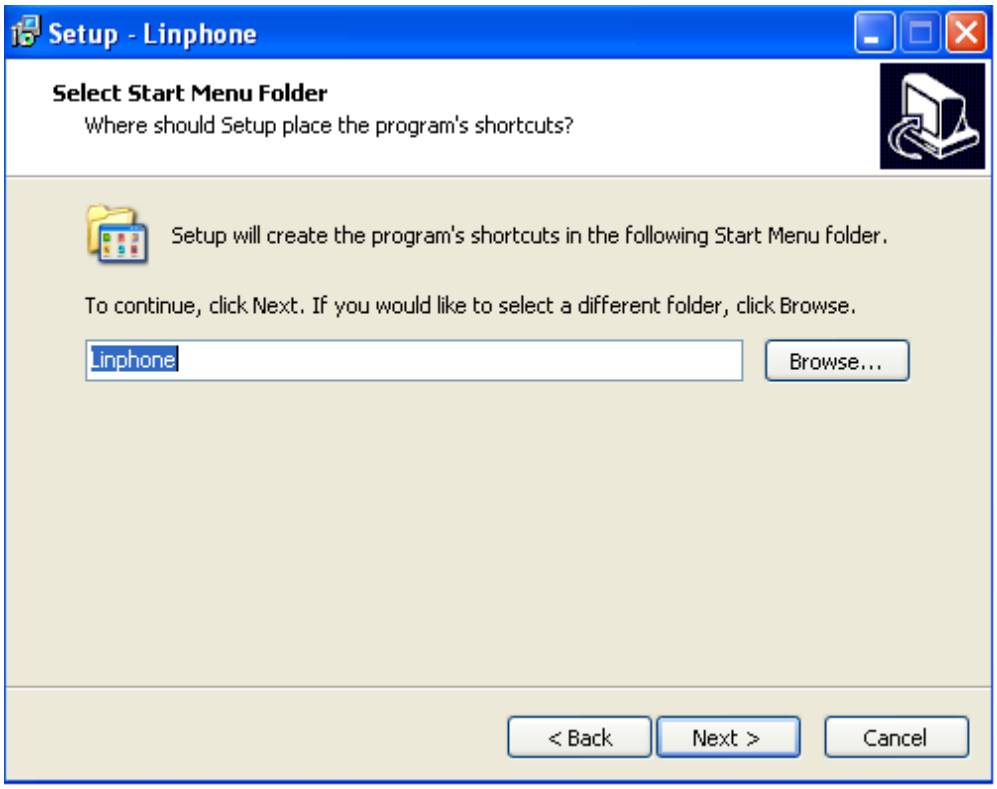
5. Tap on Next button.



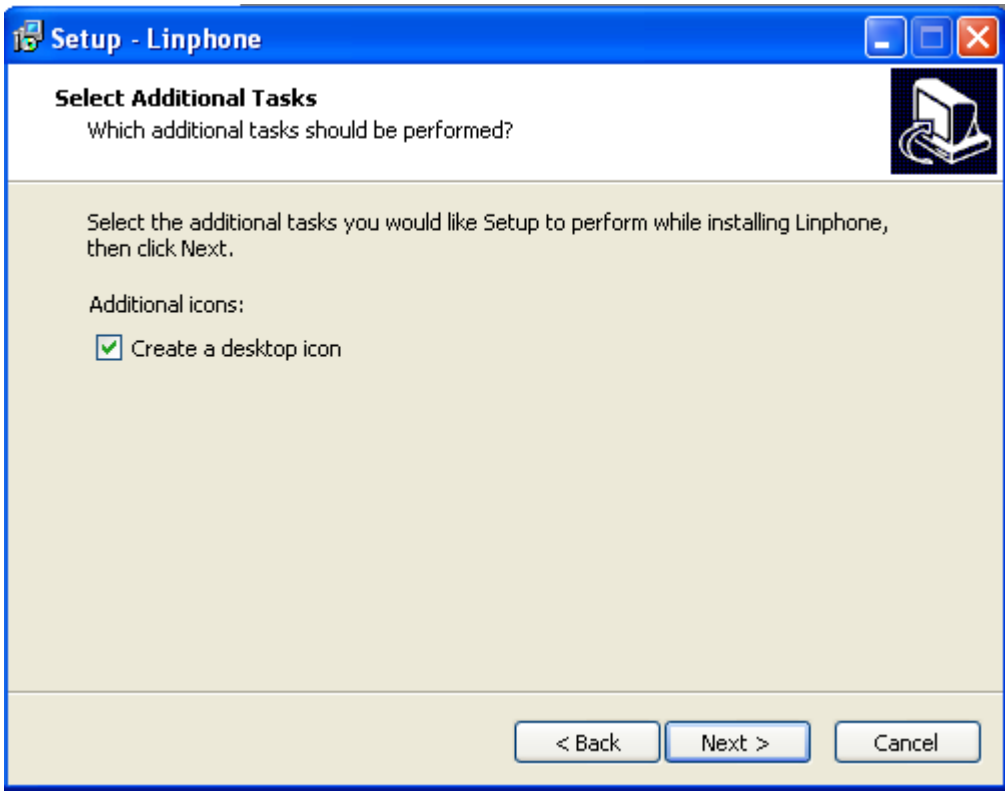
6. Tap on Next button.



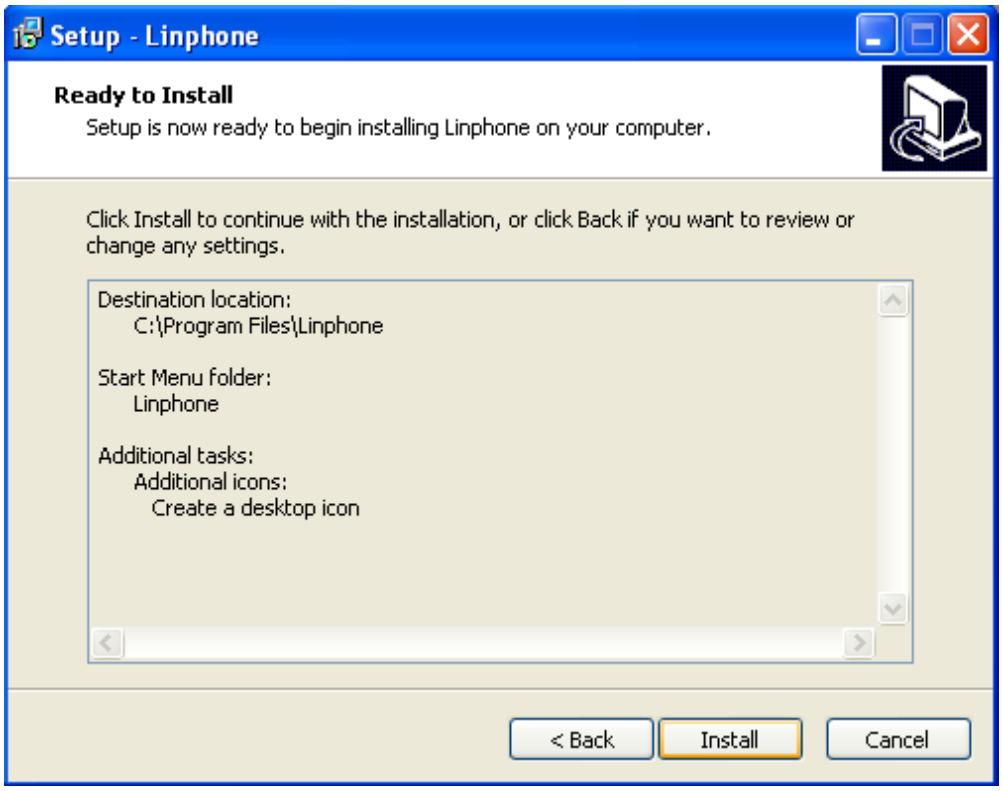
7. Tap on Next button.



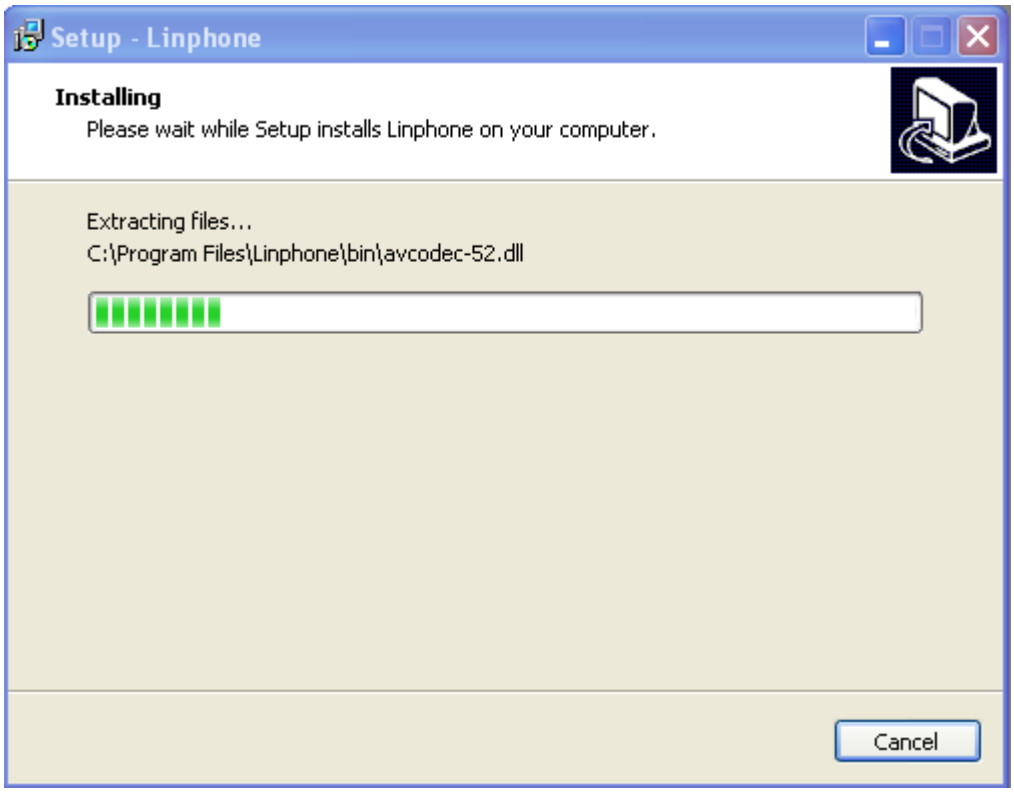
8. Tap on Next button.



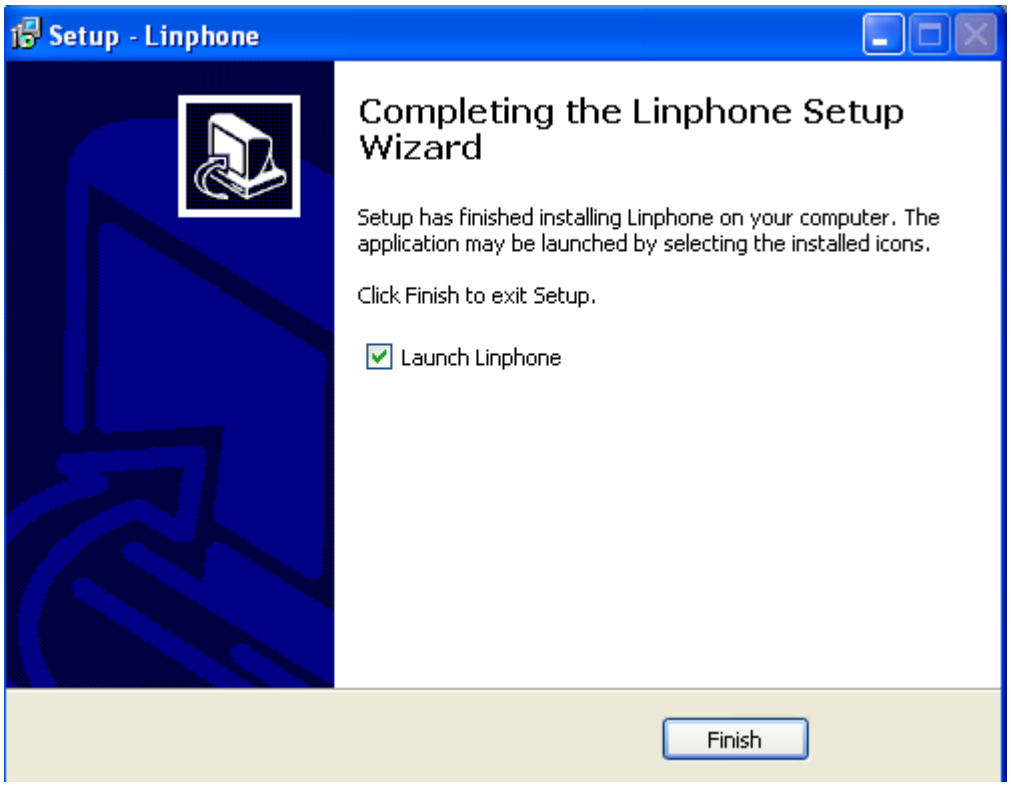
9. Tap on Next button.



10. Tap on Install button.



11. Tap on Install button to complete the installation.



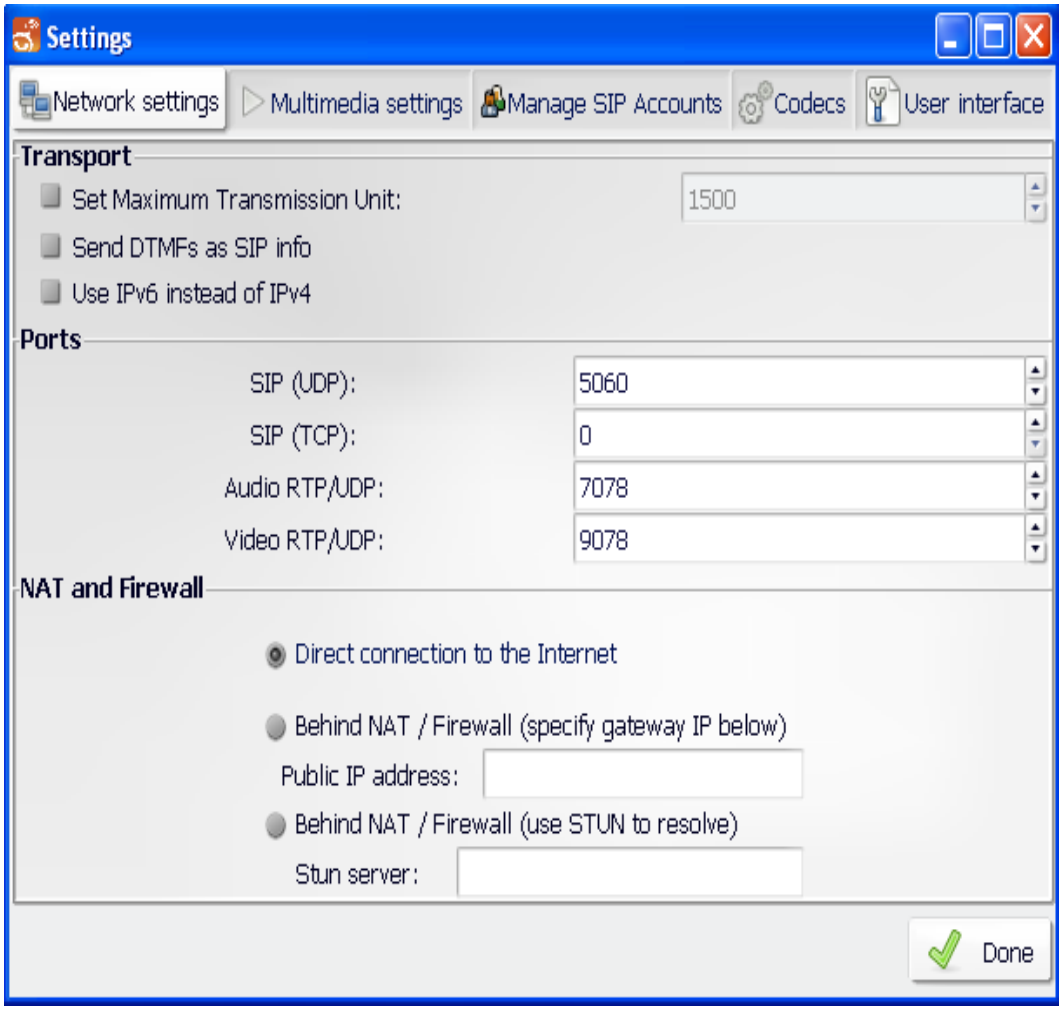
12. Linphone SIP client application will be launched.

APPENDIX D LINPHONE SIP CLIENT CONFIGURATION

This section briefly describes how to configure Linphone SIP client for laboratory exercise. The below mentioned procedure briefly explains step by step instructions to configure Linphone client for laboratory exercises (LINPHONE).

1. Launch Linphone application by clicking on Linphone icon on desktop.

2. Tap on ***Linphone*** menu option and then select ***preferences*** menu option to open settings dialog.



3. Configure TCP and UDP port fields in Network settings tab as mentioned below and remaining fields can be left with default values.

- If SIP client need to use TCP protocol as transport protocol

  SIP (UDP): 0

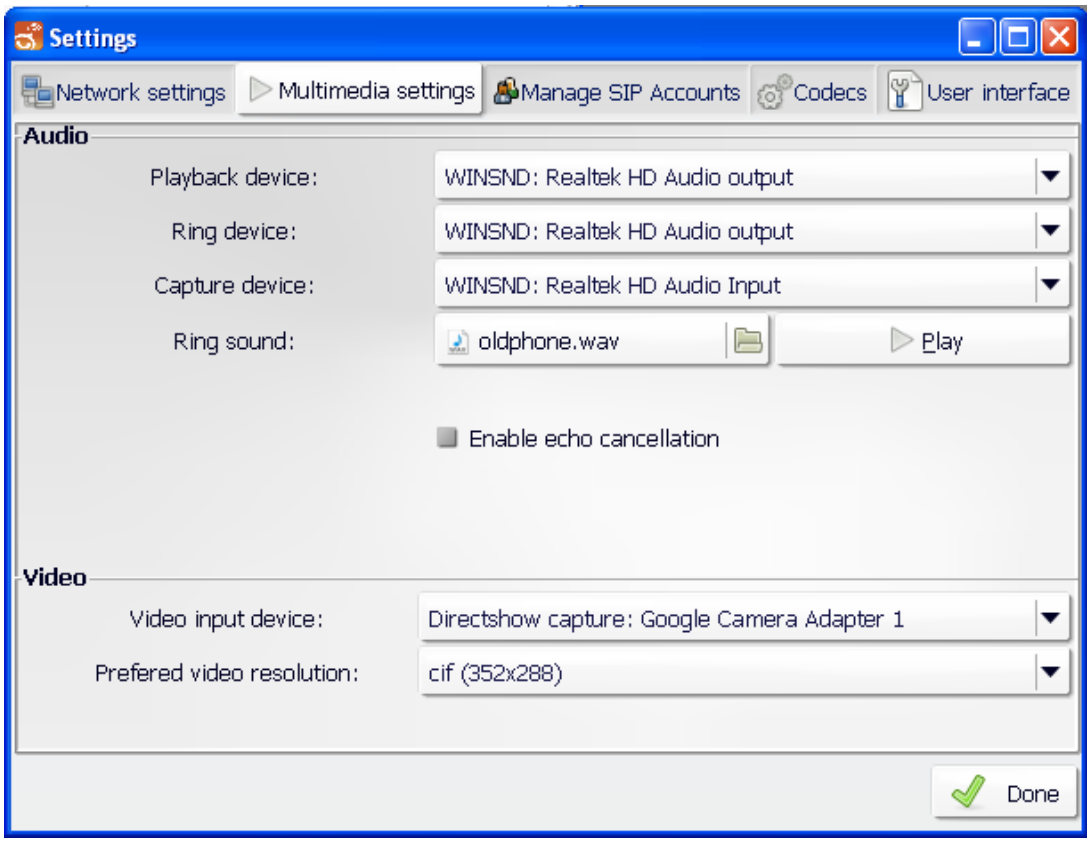
  SIP (TCP): 5060

- . If SIP client need to use UDP protocol as transport protocol

  SIP (UDP): 5060

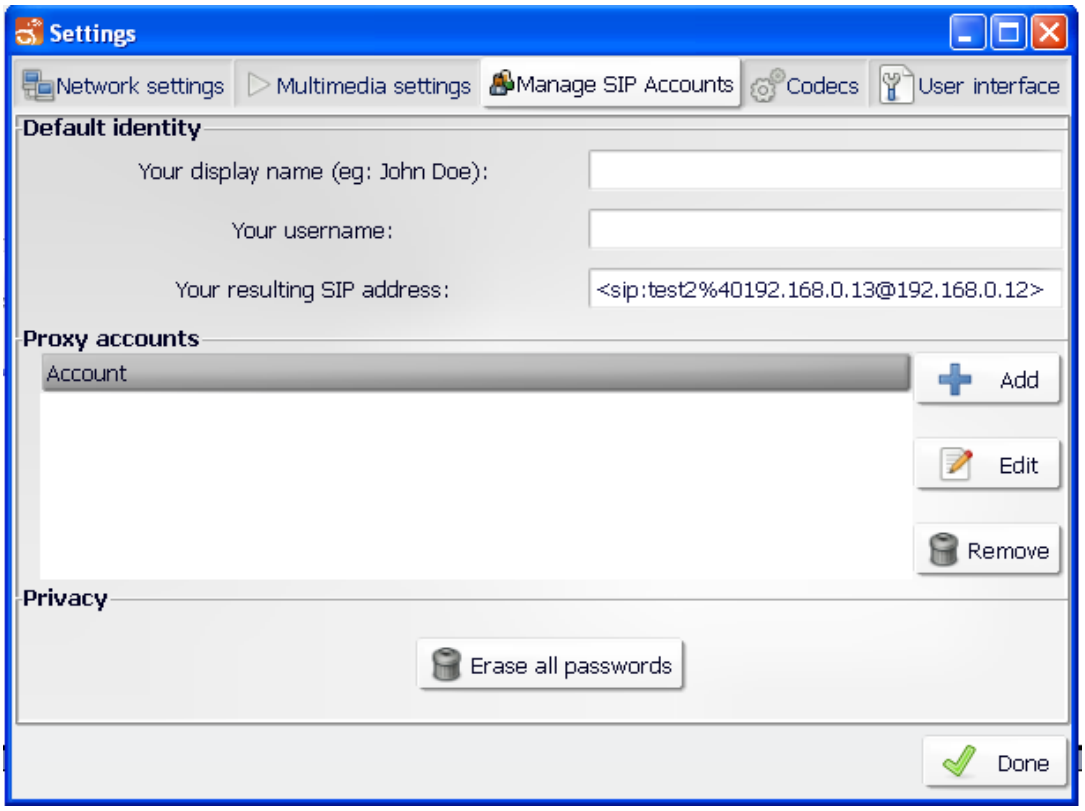
  SIP (TCP): 0

4. Click on Multimedia settings tab.



5. Configure audio and video devices i.e. (input and output devices) in Multimedia settings tab. Please make sure that selected devices are working fine. If any of the selected devices is not working it is not possible to complete all the practical lessons.

6. Tap on Manage SIP Accounts tab.



7. Configure SIP Accounts as below

   - Your display name : <username – without domain>

     Ex:-

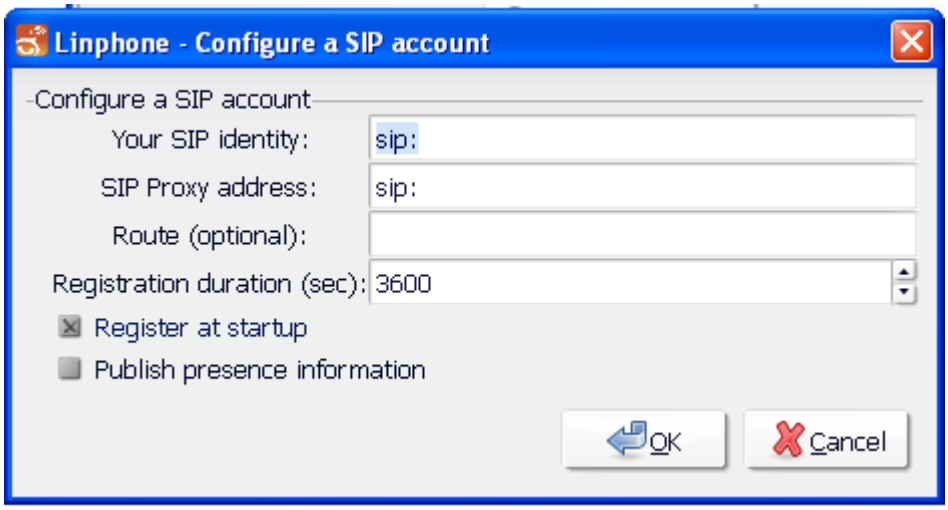
     Your display name: test2

   - Your username : <username – without domain>@<Server IP>

     Ex:-

     Your username: test2@192.168.0.13

8. Tap on Add button.



9. Configure below mentioned parameters in SIP account settings dialog and leave the remaining parameters as default values.

   - Your SIP identity: sip:<user>@<Server IP>

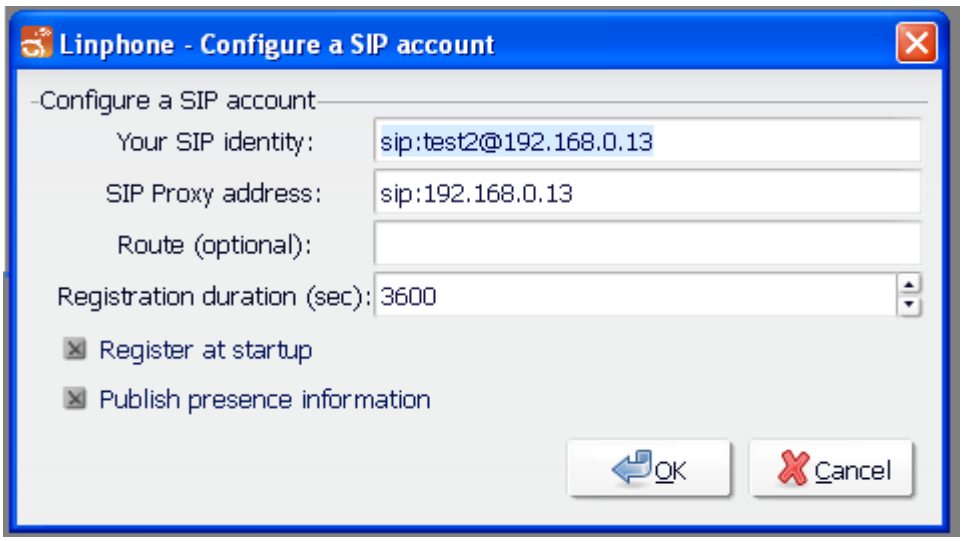
     Ex: - Your SIP identity: test2@192.168.0.13

   - SIP Proxy address: sip:<Server IP>

     Ex: - SIP proxy address: sip: 192.168.0.13

   - Check Register at startup check box.

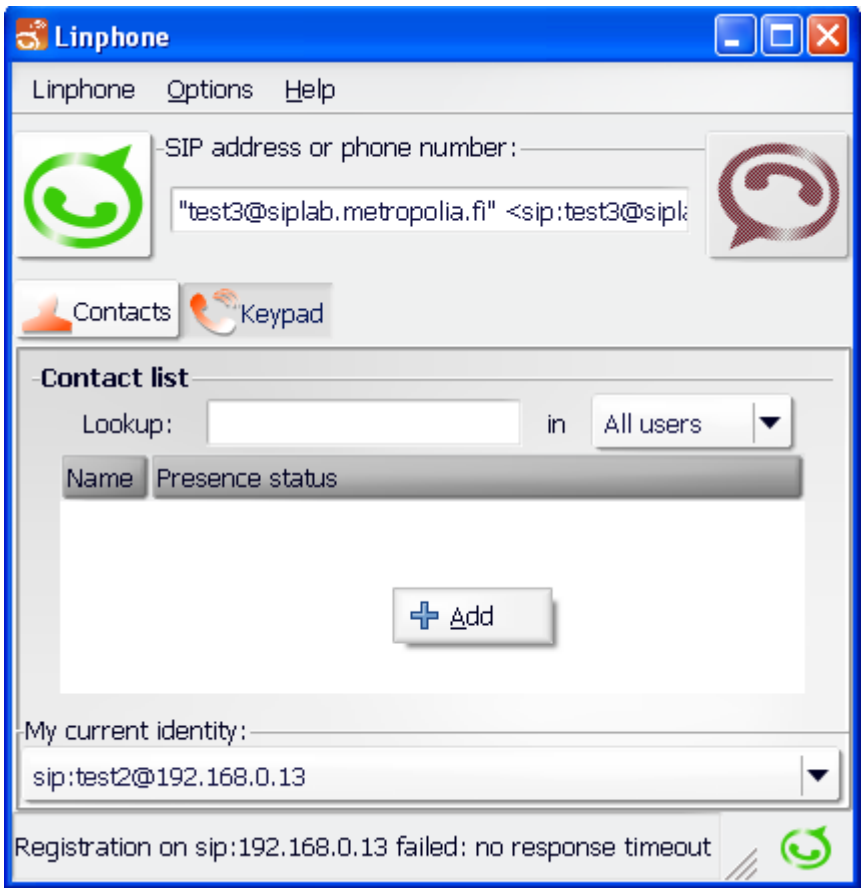
   - Check Publish presence information checkbox.



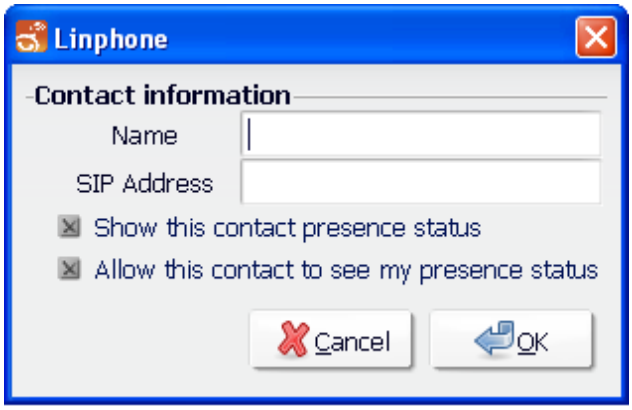
10. Tap on OK button.

11. Tap on Done button in Settings dialog.

12. Right click in the open space below presence status label.



13. Click on Add button to add new contact.



14. Configure below mentioned parameters contact information dialog.
    Please make sure that the user used in this configuration should be different than the user used in step 9. The user configured in this step should be logged in from other SIP client.

    - Name:<user>@<Server IP>

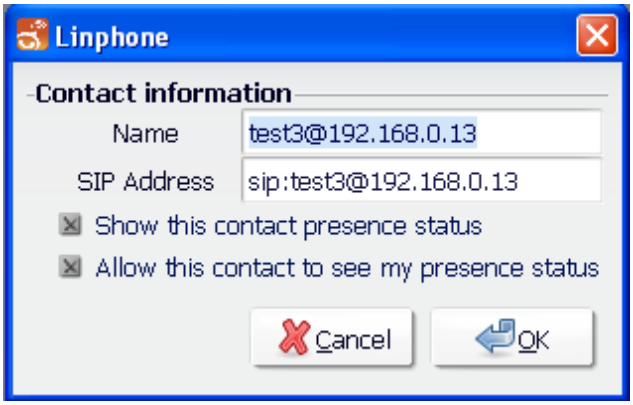
      Ex: - Name : test3@192.168.0.13

    - SIP Address: sip:<user>@<Server IP>

      Ex: - SIP Address: sip:test3@192.168.0.13

    - Check Show this contact person status check box.

    - Check Allow this contact to see my presence status check box.



15. Tap on OK button to complete the SIP client configuration for laboratory exercises.

APPENDIX E WIRESHARK NETWORK ANALYSER INSTALATION

Wireshark is the world's foremost network protocol analyzer. It lets you capture and interactively browse the traffic running on a computer network. It is the de facto (and often de jure) standard across many industries and educational institutions (WIRESHARK).

Wireshark protocol analyzer can be used for analyzing the messages exchanged between SIP client and server. It is recommended that students should install this software in the client machines along with Linphone client software. With this protocol analyzer student can monitor all the messages exchanged between client and server, which will also help in better understanding of SIP protocol functionality.

Download the latest Wireshark software from www.wireshark.org and install the same as per the instructions mentioned at the same place.